

Міністерство освіти і науки України  
Національний технічний університет України  
"Київський політехнічний інститут"

**Згуровський М. З., Бідюк П. І., Терентьев О. М., Просьянкiна-Жарова Т. І.**

## **БАЙЄСІВСЬКІ МЕРЕЖІ В СИСТЕМАХ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ**

(навчальний посібник)

Рекомендовано Методичною радою НТУУ "КПІ"

УДК (519.248: 004)(075.8)

ББК (22.172:32.97)я731

**Згуровський М. З., Бідюк П. І., Терентьев О. М., Просьянкіна-Жарова Т. І.**

**Байєсівські мережі в системах підтримки прийняття рішень**

— Київ : ТОВ «Видавниче Підприємство «Едельвейс», 2015. — 300 с.

ISBN 978-966-2748-73-4

Пропонується докладне висвітлення сучасних підходів до моделювання процесів довільної природи за допомогою байєсівських мереж. Байєсівська мережа – ймовірнісна модель, представлена у формі спрямованого ациклічного графа, вершинами якого є змінні досліджуваного процесу. Байєсівська мережа – потужний сучасний інструмент моделювання процесів та об'єктів, які функціонують в умовах наявності невизначеностей довільної природи. Їх успішно використовують для розв'язання задач прогнозування, передбачення, медичної і технічної діагностики, прийняття управлінських рішень, автоматичного керування і т. ін. Розглянуто теорію побудови байєсівських мереж, яка включає задачі навчання структури мережі та формування ймовірнісного висновку на її основі. Наведено практичні методики побудови (оцінювання) структури мережі на основі статистичних даних і експертних оцінок. Докладно описано відповідні алгоритмічні процедури. Окремо розглянуто варіанти використання дискретних і неперервних змінних, а також можливості створення гібридної мережі. Наведено кілька методів обчислення ймовірнісного висновку за допомогою побудованої мережі, у тому числі методи формування точного і наближеного висновків.

Докладно розглянуто приклади розв'язання практичних задач за допомогою мереж Байєса. Зокрема, задачі моделювання, прогнозування і розпізнавання образів. Наведено перелік відомих програмних продуктів та їх виробників для побудови та застосування байєсівських мереж, частина з яких є повністю доступними для використання у мережі Інтернет. Деякі системи можна доповнювати новими програмними модулями.

Книга рекомендується як навчальний посібник для студентів, аспірантів та викладачів, а також для інженерів, які спеціалізуються у галузі розв'язання задач ймовірнісного математичного моделювання, прогнозування, передбачення і розпізнавання образів, інформація стосовно яких представлена статистичними даними та експертними оцінками.

Рекомендовано Методичною радою НТУУ "КПІ" (протокол №10 від 19.06.2014 р.)

Ухвалено до видання Вченою радою Інституту прикладного системного аналізу НТУУ «КПІ» (протокол №7 від 22.04.2014 р.)

Рецензенти:

- *д.т.н., професор Азарков Валерій Миколайович (Національний авіаційний університет);*

- *д.т.н., професор Снитюк Віталій Євгенович (Київський національний університет імені Тараса Шевченка)*

Відповідальний редактор: *д.т.н., професор, Данілов В.Я.*

**Згуровський Михайло Захарович**

**Бідюк Петро Іванович**

**Терентьев Олександр Миколайович**

**Просьянкіна-Жарова Тетяна Іванівна**

ISBN 978-966-2748-73-4

© Згуровський М.З., Бідюк П.І., Терентьев О.М., Просьянкіна-Жарова Т.І.

## ВСТУП

Розв'язання задач математичного моделювання, прогнозування, передбачення, менеджменту, автоматичного керування та багатьох інших, як правило, виконується в умовах наявності невизначеностей довільної природи та різних типів. Невизначеності зумовлені відсутністю необхідних об'ємів інформації (експертні оцінки і статистичні дані) про досліджувані процеси, пропусками даних (вимірів), наявністю збурень станів та шумів вимірів, повністю або частково невідомою структурою об'єкта (і, відповідно, його моделі) та іншими факторами. Самими поширеними видами невизначеностей, які зустрічаються у процесі побудови математичних моделей процесів і об'єктів різної природи, є структурні, статистичні і параметричні невизначеності. Для побудови високоякісних моделей, отримання прийнятних оцінок прогнозів та відповідних їм рішень або керуючих впливів необхідно мінімізувати вплив невизначеностей шляхом використання їх можливого математичного опису або введення експертних оцінок, які забезпечать врахування невизначеності у моделях.

Сьогодні немає універсальних процедур врахування невизначеностей більшості існуючих типів, які можна успішно застосовувати при розв'язанні згаданих практичних задач. Як правило, існуючі методи обробки невизначеностей дають можливість враховувати деякі конкретні типи невизначеностей з метою підвищення якості остаточного результату. Так, наприклад, фільтр Калмана дає можливість враховувати та мінімізувати вплив збурень станів та шумів вимірів і отримати, завдяки цьому, оптимальні оцінки стану досліджуваного процесу на фоні негативних випадкових впливів. Це сприяє не тільки покращенню оцінок стану, але й підвищенню якості оцінок прогнозів стану динамічних систем і якості керування ними.

Велику допомогу у боротьбі з невизначеностями надають методи інтелектуального аналізу даних, до яких відносять нейронні мережі, нечітку логіку, нейро-нечіткі системи, байєсівські мережі та інші моделі байєсівського типу, еволюційні методи, інформований (евристичний) пошук та дослідження просторів, спрямовані на розв'язання оптимізаційних задач, математичну логіку, дерева рішень та деякі інші. Високоєфективний інструмент аналізу даних з можливим використанням експертних оцінок представляють байєсівські мережі – ймовірнісні моделі у формі спрямованих ациклічних графів, вершинами яких є змінні досліджуваних процесів.

Теорію мереж Байєса почали активно розвивати на початку 80-х років ХХ-го століття. Сьогодні їх успішно використовують для розв'язання таких практичних задач: ймовірнісне математичне моделювання процесів і об'єктів різної природи, які функціонують в умовах наявності невизначеностей, прогнозування динаміки розвитку процесів та технологічне передбачення, автоматичне діагностування в техніці та медицині, прийняття рішень у бізнесі і на виробництві, менеджмент фінансово-економічних та інших ризиків, розпізнавання образів, аналіз причинно-наслідкових зв'язків при дослідженні функціонування складних ієрархічних систем, для створення систем автоматичного керування технічними об'єктами та деяких інших задач. Успішне застосування математичних моделей у формі мереж Байєса зумовлене їх високою гнучкістю, можливістю врахування великої кількості категоріальних і числових змінних (у тому числі невимірюваних), наявністю широкого спектру методів структурного і параметричного навчання, а також формування ймовірнісного висновку у прямому і зворотному напрямках.

Мережі Байєса покладено в основу багатьох існуючих систем підтримки прийняття рішень у різних галузях діяльності людини. Потреба в таких системах підтримки прийняття рішень швидко зростає як у суто інженерних застосуваннях, так і у фінансово-економічних системах, медицині, біології, екології і т. ін. Так, системи підтримки прийняття рішень являють собою ефективний інструмент для створення математичного опису, прогнозування та керування для процесів із структурними та параметричними невизначеностями різних

типів. Вони можуть бути застосовані також до вирішення проблем передбачення при відповідній постановці та розв'язанні задачі. Сьогодні у світі успішно застосовують для розв'язання практичних задач декілька тисяч типів систем підтримки прийняття рішень різного призначення.

Перший розділ присвячено основам теорії мереж Байєса. Тут розглядаються такі поняття: граф-теоретичне представлення і типи зв'язків у мережі, стандартизований математичний запис мереж Байєса, типи мереж і теорема Байєса. Значна увага приділена їх практичному застосуванню в якості інструмент інтелектуального аналізу даних.

Другий розділ присвячено розв'язанню слабкоструктурованих задач за допомогою мереж Байєса з прихованими вершинами (змінними), як задачі системного аналізу. Розглянуто можливі випадки навчання мереж такого типу. Докладно представлено оптимізаційні процедури, які використовуються для навчання у випадку наявності прихованих вершин – градієнтний алгоритм, EM-алгоритм, структурний EM-алгоритм і алгоритм стиснення границь. Запропоновано кілька практичних прикладів побудови мережі Байєса з прихованими вершинами з використанням відомих вибірок даних.

У третьому розділі розглянуто методи обчислення ймовірнісного висновку у байєсівських мережах, зокрема алгоритми формування точного і наближеного висновків. Наведено докладний огляд цих методів, достатній для розуміння їх суті та практичного застосування.

LS-метод (метод Лорітцена-Шпігельгальтера) обчислення ймовірнісного висновку в мережах Байєса розглянуто у четвертому розділі. Наведено докладну процедуру обчислень, яка складається з двох етапів: (1) – побудова об'єднаного дерева; (2) – процедура розповсюдження повідомлень. Запропоновано приклад обчислення ймовірнісного висновку за цим методом. У сьомому розділі наведено докладні алгоритми і приклади розпізнавання символів за допомогою мереж Байєса.

Побудова (оцінювання) структури мережі Байєса розглядається у п'ятому розділі. Зокрема, наведено методи на основі оціночних функцій, методи з використанням тестів на незалежність вершин мережі та евристичні методи. Розглянуто можливості зменшення обчислювальної складності при побудові мереж – евристичний підхід, функція Купера-Герсковича, модифікована логарифмічна функція Купера-Герсковича, функція опису мінімальної довжини та інші.

Шостий розділ присвячений практичним питанням розробки систем підтримки прийняття рішень на основі байєсівських мереж.

Сьомий та восьмий розділи присвячені використанню байєсівських мереж у системах розпізнавання образів різних типів.

У дев'ятому розділі розглядаються місце та роль байєсівських мереж в технологіях інтелектуального аналізу даних. Наведено приклади застосування мереж Байєса у різних напрямках виконання науково-технічних проектів. Дано огляд програмних систем для побудови байєсівських мереж та їх виробників, вказано на можливості їх використання, зокрема детально розглянуто Застосування сучасних програмних продуктів для побудови мереж Байєса на прикладі SAS Enterprise Miner.

Висловлюємо подяку студентам-магістрам, що захистилися в 2010 році – Давиденку Володимирі Івановичу, Акулініній Олені Анатоліївні та Гуз Наталії Сергіївні за допомогу і якісно виконану роботу при виконанні обчислювальних експериментів та їх оформленні.

Автори сподіваються, що книга в цілому буде корисною для студентів, аспірантів, інженерів, викладачів та всіх інших, хто займається практичними задачами ймовірнісного математичного моделювання, прогнозування та керування на основі експериментальних і статистичних даних у різних галузях техніки, економіки, фінансів, екології, соціальних дослідженнях, біології і т. ін.

## ПЕРЕЛІК СКОРОЧЕНЬ

АКФ	– автокореляційна функція
АФК	– адаптивний фільтр Калмана
БК	– байєсівський класифікатор
НБК	– наївний байєсівський класифікатор
БМД	– байєсівська мережа довіри
ЕОМ	– електронно-обчислювальна машина
ВМВ	– упорядкована множина вершин
ВП	– випадковий процес
ДБМ	– динамічна байєсівська мережа
ЗВІ	– значення взаємної інформації
ІАД	– інтелектуальний аналіз даних
ІМ	– імітаційна модель
КГ	– функція Купера-Герсковича
МБ	– мережа Байєса
МГВА	– метод групового врахування аргументів
МКМЛ	– метод Монте-Карло для марковських ланцюгів
МЛКГ	– модифікована функція Купера-Герсковича
ММ	– математичне моделювання
ММП	– метод максимальної правдоподібності
ММС	– максимізація математичного сподівання
МНК	– метод найменших квадратів
МР	– множинна регресія
НБК	– наївний байєсівський класифікатор
НМНК	– нелінійний метод найменших квадратів
ПЗ	– програмне забезпечення
ПС	– простір станів
РММП	– рекурсивний метод максимальної правдоподібності
РМНК	– рекурсивний метод найменших квадратів
РО	– розпізнавання образів
САП	– середнє абсолютне значення похибки
СЕ	– сезонний ефект
САПП	– середня абсолютна похибка у процентах
СКП	– середньоквадратична похибка
СППР	– система підтримки прийняття рішень
ТЙ	– теорія ймовірностей
ТУІ (або ТУЙ)	– таблиця умовних ймовірностей
ТУН	– тест на умовну незалежність
УЙР	– умовний ймовірнісний розподіл
ФК	– фільтр Калмана
АІ	– artificial intelligence (штучний інтелект)
АРН	– adaptive probabilistic network (адаптивна ймовірнісна мережа)
АРА	– arc reversal algorithm (алгоритм реверсування дуг)

ARS	–	adaptive rejection sampling (адаптивна дискретизація з відхиленням)
BC-algorithm	–	bound and collapse algorithm (алгоритм стиснення границь)
BEBA	–	bucket elimination for belief assessment (алгоритм поглинаючого виключення для оцінювання довіри)
BIC	–	Bayesian information criterion (байєсівський інформаційний критерій)
BN-RAS	–	Bayesian network randomized scheme (процедура рандомізації на байєсівській мережі)
EM	–	expectation maximization (максимізація математичного сподівання)
HIS	–	heuristic importance sampling (алгоритм евристичної дискретизації за принципом важливості)
HMM	–	hidden Markov model (прихована марковська модель)
HTM	–	hierarchical temporary memory (ієрархічна тимчасова пам'ять)
IBP	–	iterative belief propagation (ітераційне розповсюдження ймовірностей)
IB-approach	–	independence-based approach (підхід, який ґрунтується на незалежності)
LS-algorithm	–	Lauritzen-Spiegelhalter algorithm (алгоритм Лорітцена-Шпігельгальтера)
MAP	–	maximum a posteriori (максимальне апостеріорне значення)
MCMC	–	Markov chain Monte Carlo (метод Монте-Карло для марковських ланцюгів)
MDL	–	minimum description length (опис мінімальної довжини)
SEM	–	structural EM-algorithm (структурний алгоритм максимізації математичного сподівання)
SIS	–	self-importance sampling (дискретизація за принципом вираженої важливості)
SPI	–	symbolic probabilistic inference (ймовірнісний висновок у символічній формі)
SS-method	–	Shenoy-Shafer method (метод Шеноя-Шафера)

ЗМІСТ

	ВСТУП .....	3
РОЗДІЛ 1	БАЙЄСІВСЬКІ МЕРЕЖІ ЯК ІНСТРУМЕНТ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ У СИСТЕМАХ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ.....	10
	1.1. Інтелектуальний аналіз даних як складова сучасних систем підтримки прийняття рішень.....	10
	1.2. Задачі, методи та алгоритми інтелектуального аналізу даних	15
	1.3. Мережа Байєса – особливий інструмент інтелектуального аналізу даних.....	20
	Контрольні питання.....	35
РОЗДІЛ 2	РОЗВ'ЯЗАННЯ СЛАБКОСТРУКТУРОВАНИХ ЗАДАЧ СИСТЕМНОГО АНАЛІЗУ ЗА ДОПОМОГОЮ БАЙЄСІВСЬКИХ МЕРЕЖ .....	37
	2.1. Застосування мереж Байєса до розв'язання задач системного аналізу.....	37
	2.2. Огляд випадків навчання мереж Байєса.....	42
	2.3. Алгоритми навчання мереж Байєса з прихованими вершинами.....	46
	2.4. Особливості побудови мережі Байєса з прихованими вершинами та невідомою структурою.....	54
	2.5. Методика знаходження параметрів байєсівських мереж з прихованими вершинами.....	55
	2.6. Приклади практичного застосування мереж Байєса з прихованими вершинами.....	57
	Контрольні питання.....	64
РОЗДІЛ 3	ЙМОВІРНІСНИЙ ВИСНОВОК У БАЙЄСІВСЬКИХ МЕРЕЖАХ.....	66
	3.1. Поняття ймовірнісного висновку у байєсівській мережі.....	66
	3.2. Алгоритми формування точного висновку.....	70
	3.3. Формування ймовірнісного висновку у байєсівській мережі на основі навчальних даних.....	95
	3.4. Методи оцінювання якості побудови ймовірнісного висновку	100
	Контрольні питання.....	100
РОЗДІЛ 4	ПОБУДОВА ЙМОВІРНІСНОГО ВИСНОВКУ ІЗ ЗАСТОСУВАННЯМ АЛГОРИТМІВ КЛАСТЕРИЗАЦІЇ.....	102
	4.1. Архітектури алгоритмів кластеризації.....	102

	4.2. Застосування <i>LS</i> – методу та його модифікацій.....	106
	4.3. Дослідження результатів роботи <i>LS</i> -методу та його модифікацій, одержаних при використанні різних програмних продуктів, призначених для побудови мереж Байєса.....	118
	4.4. Порівняння архітектури алгоритмів кластеризації.....	124
	Контрольні питання.....	129
РОЗДІЛ 5	ПОБУДОВА СТРУКТУРИ БАЙЄСІВСЬКОЇ МЕРЕЖІ .....	130
	5.1. Методи побудови структури байєсівських мереж.....	130
	5.2. Розрахункова складність задачі побудови байєсівської мережі та методи її зменшення.....	134
	5.3. Функція Купера-Герсковича.....	139
	5.4. Модифікована логарифмічна функція Купера-Герсковича.....	141
	5.5. Функція опису мінімальною довжиною.....	142
	5.6. Порівняння ефективності оціночних функцій.....	145
	5.7. Алгоритм евристичного метода побудови байєсівських мереж за навчальними даними.....	146
	5.8. Методи оцінювання якості побудови байєсівських мереж	153
	Контрольні питання.....	155
РОЗДІЛ 6	ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ МЕРЕЖ БАЙЄСА.....	157
	6.1. Загальна характеристика систем підтримки прийняття рішень.....	157
	6.2. Особливості систем підтримки прийняття рішень на основі мереж Байєса різної типології.....	160
	6.3. Система підтримки прийняття рішень для інтелектуального аналізу даних на основі байєсівських мереж.....	163
	6.4. Приклад практичної реалізації системи підтримки прийняття рішень для інтелектуального аналізу даних на основі мереж Байєса.....	166
	Контрольні питання.....	173
РОЗДІЛ 7	МЕРЕЖІ БАЙЄСА ЯК ІНСТРУМЕНТ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ОБРАЗІВ.....	174
	7.1. Задача розпізнавання образів.....	174
	7.2. Методи розпізнавання образів.....	176
	7.3. Класифікація методів розпізнавання образів.....	177
	7.4. Застосування систем розпізнавання образів.....	182
	7.5. Системи розпізнавання образів на основі наївного байєсівського класифікатора.....	184
	7.6. Динамічна мережа Байєса.....	187



	Контрольні питання.....	196
РОЗДІЛ 8	СИСТЕМА РОЗПІЗНАВАННЯ СИМВОЛІВ НА ОСНОВІ ДИНАМІЧНОЇ МЕРЕЖІ БАЙЄСА.....	197
8.1.	Опис алгоритму розпізнавання текстових символів на основі мереж Байєса.....	197
8.2.	Приклад побудови структури програмної реалізації системи розпізнавання символів на основі мереж Байєса.....	199
8.3.	Емпіричне оцінювання алгоритмів розпізнавання.....	201
8.4.	Огляд комп'ютерних систем для розпізнавання символів.....	210
	Контрольні питання.....	214
РОЗДІЛ 9	СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ МОДЕЛЮВАННІ ПРОЦЕСІВ РІЗНОЇ ПРИРОДИ ТА ЇХ ПРОГРАМНА РЕАЛІЗАЦІЯ.....	215
9.1.	Огляд задач, які успішно вирішуються із застосуванням байєсівських мереж.....	215
9.2.	Огляд комп'ютерних програм для створення та застосування мереж Байєса.....	228
9.3.	Застосування сучасних програмних продуктів для побудови мереж Байєса на прикладі SAS Enterprise Miner .....	230
9.4.	Приклади побудови систем підтримки прийняття рішень із використанням мереж Байєса.....	257
	Контрольні питання.....	263
	ЗАВДАННЯ ДЛЯ ВИКОНАННЯ ПРАКТИЧНИХ РОБІТ.....	264
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	275
	ДОДАТКИ.....	292

## РОЗДІЛ 1

### БАЙЄСІВСЬКІ МЕРЕЖІ ЯК ІНСТРУМЕНТ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ У СИСТЕМАХ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

- Інтелектуальний аналіз даних як складова сучасних систем підтримки прийняття рішень
- Задачі, методи та алгоритми інтелектуального аналізу даних
- Мережа Байєса як інструмент інтелектуального аналізу даних
- Основні поняття теорії байєсівських мереж

#### 1.1. Інтелектуальний аналіз даних як складова сучасних систем підтримки прийняття рішень

З кожним роком стрімко зростає інформаційна лавина, що надходить з науки, бізнесу, соціальних мереж та інших джерел. На перший погляд вичерпну інформацію про будь-який об'єкт можна знайти просто і швидко, однак для виявлення дійсно корисної інформації, навіть експерту в певній галузі, необхідно докласти чимало зусиль, адже він повинен не лише проаналізувати великі масиви інформації, визначитися із вибором оптимальних для вирішення конкретної задачі методів та алгоритмів, виходячи як із своїх знань і досвіду, так і знань, що містяться у даних, що аналізуються.

Тому проблема, коли в процесі аналізу даних різної природи дослідник зустрічається з так званими невизначеностями, тобто неповнотою інформації про досліджуваний процес, значними впливами випадкових збурень і шумів (похибок) вимірів, нестаціонарною структурою і параметрами процесів [1, 2], стає дедалі актуальнішою. Тобто можна говорити про наявність структурних, статистичних і параметричних невизначеностей, які ускладнюють процес аналізу даних і погіршують якість остаточного результату — оцінок прогнозів та/або якості керування (автоматизованого або автоматичного).

Досить ефективним інструментом оцінювання і врахування невизначеностей статистичного характеру є, наприклад, адаптивний фільтр Калмана (ФК), який дає можливість оцінювати і прогнозувати стан динамічних процесів [2—5] у реальному часі. Адаптація моделі до характеристик випадкових збурень і шумів вимірів досягається в даному випадку використанням обчислених у реальному часі оцінок коваріаційних матриць зазначених випадкових процесів. Перевагами процедур оптимальної фільтрації є можливість врахування у явному вигляді статистичних характеристик збурень стану і шумів вимірів, обчислення оптимальних оцінок змінних стану та їх прогнозів, можливість оцінювання невимірюваних компонентів вектора стану та одночасне оцінювання станів і деяких параметрів моделі. До недоліків можна віднести значне зниження якості оцінок прогнозів у випадку, коли кількість кроків прогнозування більше одного, чутливість процедури оцінювання станів до ступеня адекватності моделі та значне ускладнення обчислювальних алгоритмів у випадку аналізу нелінійних процесів.

Значних успіхів досягнуто у напрямку моделювання і прогнозування процесів з невизначеностями за допомогою нечіткої логіки та нейронечітких методів [6, 7]. При використанні методів цього класу невизначеності відносять до "нечітких"

лінгвістичних значень змінних і правил прийняття рішень. Процедури моделювання на основі нечіткої логіки характеризуються відносною прозорістю (простотою) та можливістю їх адаптації до процесів визначеного класу. До недоліків можна віднести необхідність генерування великої кількості правил при дослідженні багатовимірних процесів і неможливість стеження за їх використанням особою, що приймає рішення (ОПР) при формуванні остаточних висновків.

Ще один широкий клас методів моделювання, прогнозування і керування, які також спрямовуються на боротьбу з невизначеностями, ґрунтується на байєсівському підході [8— 12].

Використання байєсових мереж для аналізу процесів різної природи, діяльності людини та функціонування технічних систем дозволяє враховувати та використовувати будь-які вхідні дані – експертні оцінки і статистичну інформацію. Змінні, що використовуються у мережах, можуть бути як дискретними, так і неперервними, а характер їх надходження при аналізі та прийнятті рішення може бути і в режимі реального часу, і у вигляді статистичних масивів інформації та баз даних. Завдяки представленню взаємодії між факторами процесу у вигляді причинно-наслідкових зв'язків в мережі досягається максимально високий рівень візуалізації та, як наслідок, чітке розуміння суті взаємодії факторів процесу між собою. Саме це відрізняє мережі Байєса від інших методів інтелектуального аналізу даних. Також перевагами мереж Байєса є можливості врахування невизначеностей статистичного, структурного і параметричного характеру, побудова моделей за наявності прихованих вершин і при неповних спостереженнях, а також формування висновку за допомогою різних методів – наближених і точних. Загалом можна сказати, що мережі Байєса – це високоресурсний метод ймовірнісного моделювання процесів довільної природи з невизначеностями різних типів, який забезпечує можливість достатньо точного опису їх функціонування, оцінювання прогнозів та побудову системи управління.

Методологія байєсівського аналізу даних та експертних оцінок цілком узгоджується з логікою дій особи, що приймає рішення при аналізі процесів довільної природи, формуванні альтернатив і прийнятті рішень. Апріорна інформація про досліджуваний процес доповнюється даними експерименту, додатковою інформацією якісного або кількісного характеру, що може бути отримана з різних джерел, і на основі інтегрованих знань і даних формується апостеріорний ймовірнісний висновок стосовно змінних, параметрів, станів, ситуацій та ін. Байєсівські методи успішно застосовуються на всіх етапах аналізу даних при моделюванні, прогнозуванні та прийнятті рішень. На етапі попередньої обробки даних застосовують ймовірнісну фільтрацію спостережень, заповнення пропусків, на етапі моделювання — формування структур моделей і оцінювання їх параметрів, а на етапі формування альтернатив — обчислення ймовірнісних висновків (рішень) за допомогою побудованих раніше моделей. Байєсівські методи мають такі переваги: можливість урахування невизначеностей статистичного, структурного і параметричного характеру, поєднання в одній моделі (наприклад, у байєсівській мережі) великої кількості різнорідних змінних, наявність досить гнучких процедур оцінювання параметрів і станів досліджуваних процесів, а також наявність широкого спектра методів формування точних і наближених висновків. До недоліків можна віднести труднощі з отриманням апріорної інформації та відносну складність деяких

обчислювальних процедур, пов'язаних з числовим інтегруванням, оцінюванням параметрів і формуванням імовірнісних висновків. Стосовно недоліків можна сказати, що в деяких випадках вони дійсно існують і створюють труднощі для дослідника, але з набуттям досвіду використання цих методів та підвищенням якості відповідних обчислювальних процедур аналізу даних і знань рівень та обсяг цих труднощів істотно зменшується [8].

Зважаючи на необхідність підвищення ефективності боротьби з невизначеностями та наявністю досить широкого спектра методів байєсівського аналізу даних, необхідно знати можливості їх практичного застосування, переваги та недоліки. Це особливо важливо в контексті створення систем підтримки прийняття рішень (СППР) [13—15], оскільки ймовірнісні методи і моделі дають можливість отримати важливі додаткові альтернативи для методів прийняття рішень на основі регресійного аналізу, нечіткої логіки, нейронних мереж та ін.

Завданням систем підтримки прийняття рішень є виключно надання користувачеві допомоги у тому, щоб знайти саме те рішення, яке він вважає оптимальним. Але знайти рішення, що є виключно оптимальним, самостійно користувач не може, оскільки задачі такого класу характеризуються великою розмірністю та складністю.

Такі проблеми виникають у складних технологічних, економічних, соціальних, екологічних і біологічних системах, вони характеризуються умовами невизначеності різноманітної природи, наявністю неповної, нечіткої інформації та ускладнюються впливом людського фактору.

Крім того, традиційні підходи до прийняття рішень, в основу яких був покладений виключно дедуктивний підхід з використанням правил логічного виводу, типу "що-якщо", передбачали, що досліджувана проблема буде формалізованою. Однак існує багато слабоформалізованих задач, для яких складно, але можливо знайти оптимальне рішення, а крім того, є задачі, для яких взагалі неможливо знайти оптимальне рішення традиційними методами, основанийими виключно на формалізованих методах, тому особа, що приймає рішення, повинна спиратися на існуючі прецеденти, адаптуючи варіанти рішення до конкретної ситуації. Особливістю процесу прийняття рішень є його суб'єктивний характер, користувач приймає рішення виключно на власний розсуд, користуючись при цьому як власними знаннями, так і знаннями експертів тощо. Перші експертні системи в процесі підготовки прийняття рішення переважно використовували дедуктивний вивід, оснований на правилах виду "якщо-то", які на основі вхідних даних формували експертний висновок. Однак такий варіант побудови висновку є оптимальним для формалізованих задач, для яких розроблені відповідні методи розв'язання, що мають опрацьоване математичне підґрунтя. Значна кількість задач прийняття рішень є слабо формалізованими, крім того, існує велика група задач, для яких взагалі неможливо знайти формалізоване рішення. І тому все актуальнішою стає проблема знайти хоча б одне прийнятне з множини рішень у випадках, коли неможливо знайти оптимальне або неможливо виявити всі варіанти рішення. У таких випадках краще застосовувати технології логічного висновку, засновані на прецедентах, тобто використанні попередніх знань експертів про рішення, що були застосовані в аналогічних ситуаціях. По-іншому, даний метод являє собою спосіб вирішення задач прийняття рішень на основі пошуку аналогій,

що зберігаються в базі знань системи. Пошук прецедентів та вибір оптимального рішення з набору аналогій потребують поглибленого дослідження предметної області.

Розв'язання задач такого типу належить до кола завдань системного аналізу з одного боку і задач теорії прийняття рішень з іншого. Необхідність застосування різних підходів до розв'язання формалізованих та слабкоформалізованих задач враховуючи при цьому невизначеності, вимагає створення нових підходів і методів до моделювання, прогнозування і керування в умовах наявності згаданих невизначеностей, спонукає до створення сучасних високоефективних систем підтримки прийняття рішень.

Системи підтримки прийняття рішень розпочали свій розвиток ще у 60-х роках ХХ століття із систем управлінських рішень (Management Decision Systems), призначених для обслуговування менеджерів великих компаній. Сьогодні це потужні програмні системи, що дозволяють вирішувати неформалізовані та слабкоформалізовані задачі.

Системи підтримки прийняття рішень (Decision Support System) сьогодні широко застосовуються у всіх сферах, де є необхідність прийняття рішень на основі аналізу даних: економіка та бізнес, фінанси, техніка, біологія, екологія та медицина тощо.

Залежно від класу вирішуваних задач, можна визначити такі **типи систем підтримки прийняття рішень**:

- експертні системи в медицині, техніці, економіці, фінансах, наукових дослідженнях;
- на основі геоінформаційних систем;
- використовувані для оптимального управління об'єктами різних типів в різних галузях;
- системи автоматизованого проектування (САПР);
- оптимізаційного типу для знаходження оптимальних розв'язків задач методами дослідження операцій або оптимального керування;
- для аналізу якості продукції та вироблення рекомендацій щодо підвищення якості;
- призначені для аналізу особистих проблем.

В спеціальній літературі можна знайти багато означень систем підтримки прийняття рішень, наведемо деякі з них, найбільш поширені в Україні та країнах СНД.

Система підтримки прийняття рішень – комп'ютерна інформаційна система, яка надає можливість розв'язання задач моделювання, прогнозування та/або оптимального керування кількома альтернативними методами, а також містить множину критеріїв вибору кращого рішення. Така система обов'язково містить метод прийняття рішень у явному вигляді.

Більш загальним (прийнятим за кордоном) означенням є таке.

**Система підтримки прийняття рішень** – це комп'ютерна інформаційна система, яка надає користувачу будь-яку допомогу при прийнятті рішень, наприклад, пошуку інформації, статистичної обробки даних, розв'язання оптимізаційних задач, обчислень різного характеру і т. ін.

Або інший варіант означення: система підтримки прийняття рішень –

інтерактивна автоматизована система, яка допомагає особі, яка приймає рішення, використовувати дані і моделі для розв'язання слабо структурованих і неструктурованих задач.

Всі наведені означення зосереджують увагу на тому, що системи підтримки прийняття рішень, незважаючи на своє різноманіття, призначені не для повної автоматизації процесу прийняття рішень, а лише для надання допомоги особі, що приймає рішення, у вигляді різних варіантів обробки даних та знань, побудови сценаріїв тощо.

Загалом, слід відзначити мільтидисциплінарність систем підтримки прийняття рішень, оскільки їх невід'ємними складовими є бази даних і бази знань, засоби штучного інтелекту, різноманітні методи імітаційного та математичного моделювання тощо. Задачі аналізу даних стають дедалі актуальнішими, особливо в сучасних умовах, коли накопичено значні обсяги інформації практично з усіх напрямів діяльності людства.

Вирішення цієї задачі безпосередньо пов'язане з накопиченням інформації щодо певної предметної області, зберіганням та аналізом даних. Тому на перший план у системах підтримки прийняття рішень виходять саме ті складові, що забезпечують обробку даних. А. А. Барсегян, М.С. Куприянов та ін. виділяють такі задачі обробки даних [18], як інформаційно-пошукові, оперативно-аналітичні та інтелектуальні. Вирішення цих задач потребує застосування широкого спектру методів підтримки прийняття рішень, який може доповнюватись зацікавленими особами, у тому числі і за рахунок доповнень, створюваних користувачем із використанням вбудованих засобів програмування. Тобто складність проблеми пов'язана як з необхідністю обробки значних масивів багатовимірних та різноманітних даних, від якості та своєчасності якої в значній мірі залежить оптимальність прийнятого рішення, так і від рівня забезпеченості особи, що приймає рішення, оптимальним інструментарієм для аналізу даних. Тому на особливу увагу заслуговують технології, що реалізують інтелектуальний аналіз даних (data-mining).

**Інтелектуальний аналіз даних** – мультидисциплінарна область, що виникла та розвивається на основі таких наук, як прикладна статистика, розпізнавання образів, штучний інтелект, теорія баз даних та ін. (рис. 1.1).

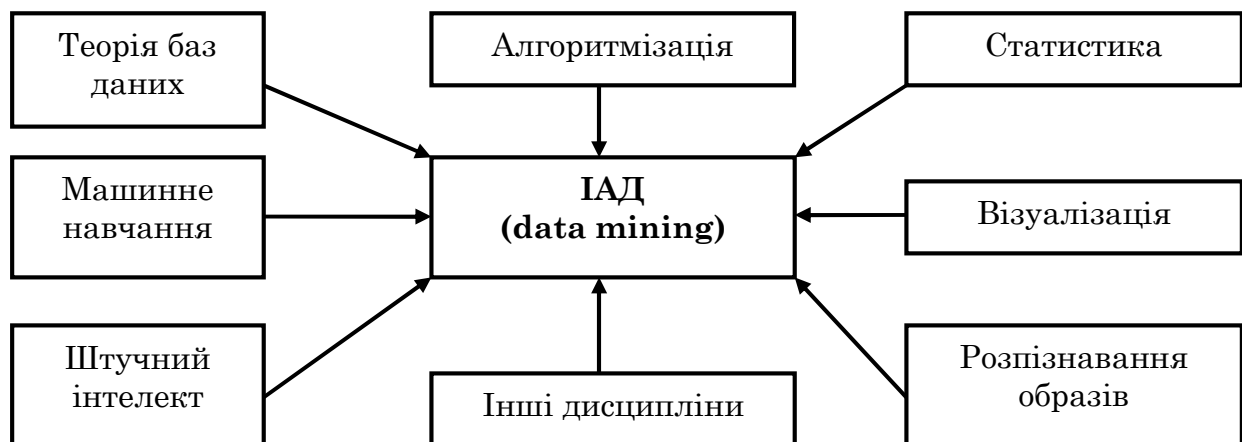


Рис. 1.1. Інтелектуальний аналіз даних як мультидисциплінарна область [20]

Наведемо короткий опис деяких дисциплін, на перетині яких з'явилася технологія інтелектуальний аналіз даних.

**Статистика** – це наука про методи збору даних, їх обробки та аналізу для виявлення закономірностей, наявних у досліджуваному явищі. Статистика складається з методів планування експерименту, збору, представлення та узагальнення даних, а також аналізу та формування висновків на підставі цих даних. Статистика оперує даними, отриманими в результаті спостережень або експериментів.

**Машинне навчання** – це наука, що вивчає комп'ютерні алгоритми, які автоматично поліпшуються під час їх застосування. Машинне навчання можна охарактеризувати як процес одержання програмою нових знань. Один із найбільш популярних прикладів алгоритму машинного навчання представляють нейронні мережі. Зокрема, теорія байєсівських мереж також належить до наукового напрямку машинного навчання.

**Штучний інтелект** – науковий напрям, у межах якого формулюються та вирішуються завдання апаратного або програмного моделювання видів людської діяльності, що традиційно вважаються інтелектуальними. Термін інтелект походить від латинського *intellectus*, що означає розум, інтелектуальні здібності людини. Відповідно, штучний інтелект (AI –artificial intelligence) тлумачиться як властивість автоматичних систем брати на себе окремі функції інтелекту людини. Штучним інтелектом називають властивість інтелектуальних систем виконувати творчі функції, які традиційно вважаються прерогативою людини.

### 1.2. Задачі, методи та алгоритми інтелектуального аналізу даних

Серед методів та алгоритмів інтелектуального аналізу даних вирізняють штучні нейронні мережі, дерева рішень, символні правила, методи найближчого сусіда і  $k$  – найближчого сусіда, метод опорних векторів, байєсівські мережі, лінійну регресію, кореляційно-регресійний аналіз; ієрархічні методи кластерного аналізу, неієрархічні методи кластерного аналізу, у тому числі алгоритми  $k$  – середніх і  $k$  – медіани; методи пошуку асоціативних правил, у тому числі алгоритм Apriori; метод обмеженого перебору, еволюційне програмування і генетичні алгоритми, різноманітні методи візуалізації даних та інші методи. Уточнюючи застосовану термінологію, слід відзначити наступне.

**Метод** представляє собою спосіб або правило, певний шлях, прийняття рішень теоретичного, практичного, пізнавального, управлінського характеру.

**Алгоритм** – це точний припис послідовності дій (кроків), які перетворюють вхідні дані у потрібний результат.

Найкращими, за рейтингом міжнародної щорічної конференції IEEE-ICDM (International Conference on Data Mining), визначено десять алгоритмів інтелектуального аналізу даних (табл. 1.1.) [19], їх розробники відзначені такими престижними преміями та нагородами, як ACMKDD (Innovation Award), IEEEICDM (Research Contributions Award) та Google Scholar.

Таблиця 1.1

Десять кращих алгоритмів для інтелектуального аналізу даних за рейтингом конференції IEEE-ICDM

№	Назва алгоритму	Група методів
1	<b>C 4.5</b>	дерева рішень
2	<b><math>k</math> – середніх</b>	неієрархічного кластерного аналізу
3	<b>Support Vector Machine (SVM)</b> або метод опорних векторів	класифікації
4	<b>Apriori</b>	пошуку асоціативних правил
5	<b>EM</b> (expectation maximization) або максимізації математичного сподівання	кластерного аналізу
6	<b>PageRank</b>	пошукових Web-алгоритмів
7	<b>AdaBoost</b> або посилення слабких класифікаторів	класифікації
8	<b>kNN</b> ( $k$ – nearest neighbor classification) або метод $k$ – найближчого сусіда	класифікації
9	<b>NaiveBayes</b>	байєсівські мережі
10	<b>CART</b> (Classification and Regression Trees)	дерева рішень

Представлені в таблиці 1.1 алгоритми застосовуються до розв'язання широкого кола різноманітних задач інтелектуального аналізу даних.

Найбільш поширеними є такі:

**1. Класифікація.** Найпростіше і найбільш розповсюджене завдання інтелектуального аналізу даних, метою якого є виявлення ознак, які характеризують групи об'єктів досліджуваного набору даних – класи; по цих ознаках новий об'єкт можна віднести до того або іншого класу. Для розв'язання задачі класифікації використовують такі методи: найближчого сусіда;  $k$  – найближчого сусіда; байєсівські мережі; дерева рішень; нейронні мережі.

**2. Кластеризація.** Кластеризація – це логічне продовження ідеї класифікації. Це завдання складніше, особливість кластеризації полягає у тому, що на початку класи об'єктів не визначені. Результатом кластеризації є розділення об'єктів на групи. Для розв'язання цієї задачі використовують карти Кохонена, байєсівські мережі та інші методи.

**3. Асоціація.** Виконується пошук закономірностей між зв'язаними подіями в наборі даних. На відміну від двох попередніх завдань інтелектуального аналізу даних, пошук закономірностей здійснюється не на основі властивостей аналізованого об'єкта, а між декількома подіями, які відбуваються одночасно. Найбільш відомий алгоритм розв'язання задачі пошуку асоціативних правил – алгоритм Apriori.



**4. Послідовна асоціація або просто послідовність.** Послідовність дає можливість знайти тимчасові закономірності між транзакціями. Завдання послідовності подібне асоціації, але її метою є встановлення закономірностей не між одночасно наступаючими подіями, а між подіями, зв'язаними у часі (тобто у часі, що відбуваються з деяким певним інтервалом). Інакше кажучи, послідовність визначається високою ймовірністю ланцюжка зв'язаних у часі подій. Наприклад, після придбання квартири мешканці у 60% випадків протягом двох тижнів купляють холодильник, а протягом двох місяців у 50% випадків купляють телевізор.

**5. Прогнозування.** У результаті розв'язання задачі прогнозування на основі особливостей історичних даних оцінюють пропущені або майбутні значення цільових чисельних показників. Для розв'язання таких задач широко застосовують методи математичної статистики та нейронні мережі.

**6. Визначення відхилень або викидів.** Мета розв'язання цієї задачі – виявлення та аналіз даних, які найбільше відрізняються від загальної множини даних, виявлення так званих нехарактерних шаблонів.

**7. Оцінювання.** Задача оцінювання зводиться до прогнозування неперервних значень ознаки.

**8. Аналіз зв'язків.** Полягає в пошуку залежностей у наборах даних.

**9. Візуалізація.** У результаті візуалізації створюється графічний образ аналізованих даних. Для розв'язання цієї задачі використовують графічні методи, що показують наявність закономірностей у даних.

**10. Підведення підсумків.** Мета цієї задачі – опис конкретних груп об'єктів з аналізованого набору даних.

Для того, щоб максимально використати можливості масштабованих інструментів інтелектуального аналізу даних комерційного рівня, необхідно вибрати, очистити та перетворити дані, іноді інтегрувати інформацію, добуту із зовнішніх джерел, і створити спеціальне середовище для роботи алгоритмів інтелектуального аналізу даних. Результати інтелектуального аналізу даних переважно залежать від рівня підготовки даних, а не від “чудодійних можливостей” якогось алгоритму або множини алгоритмів. Адже, за даними експертів [20], близько 75% робіт з інтелектуального аналізу даних припадає саме на їх збір.

Перед використанням певної технології інтелектуального аналізу даних необхідно ретельно проаналізувати її проблеми, обмеження та критичні питання, які зв'язані з нею, а також зрозуміти, що ця технологія не може надати. Навіть найдосконаліша технологія інтелектуального аналізу даних не може замінити аналітика, адже сама по собі технологія не може дати відповіді на ті питання, які не були задані. Вона лише надає аналітику потужний інструмент для полегшення і поліпшення його роботи.

Оскільки інтелектуальний аналіз даних – це мультидисциплінарна область, то для розробки програмного забезпечення, що включає засоби інтелектуального аналізу даних, необхідно задіяти фахівців з різних областей, а також забезпечити їх високоякісну взаємодію. Неможливо видобувати корисну інформацію без розуміння суті даних, тому використання технологій інтелектуального аналізу даних має бути нерозривно пов'язане з високою кваліфікацією користувача.

Необхідно підкреслити, що не всі технології можуть досягти етапів піку очікувань і подолання недоликів, передчасно завершуючи свій цикл без перспектив

до продуктивного використання. Компанія Gartner [21] ще у 1995 році запропонувала термін “Цикл зрілості технології” (англійською – Hypecycle). Суть поняття полягає в тому, що кожна технологічна інновація проходить декілька етапів для досягнення зрілості, при цьому кожен з них характеризується різними рівнями зацікавленості професіоналів та суспільства. На рис. 1.2 наведено відповідні етапи реалізації такої технології.

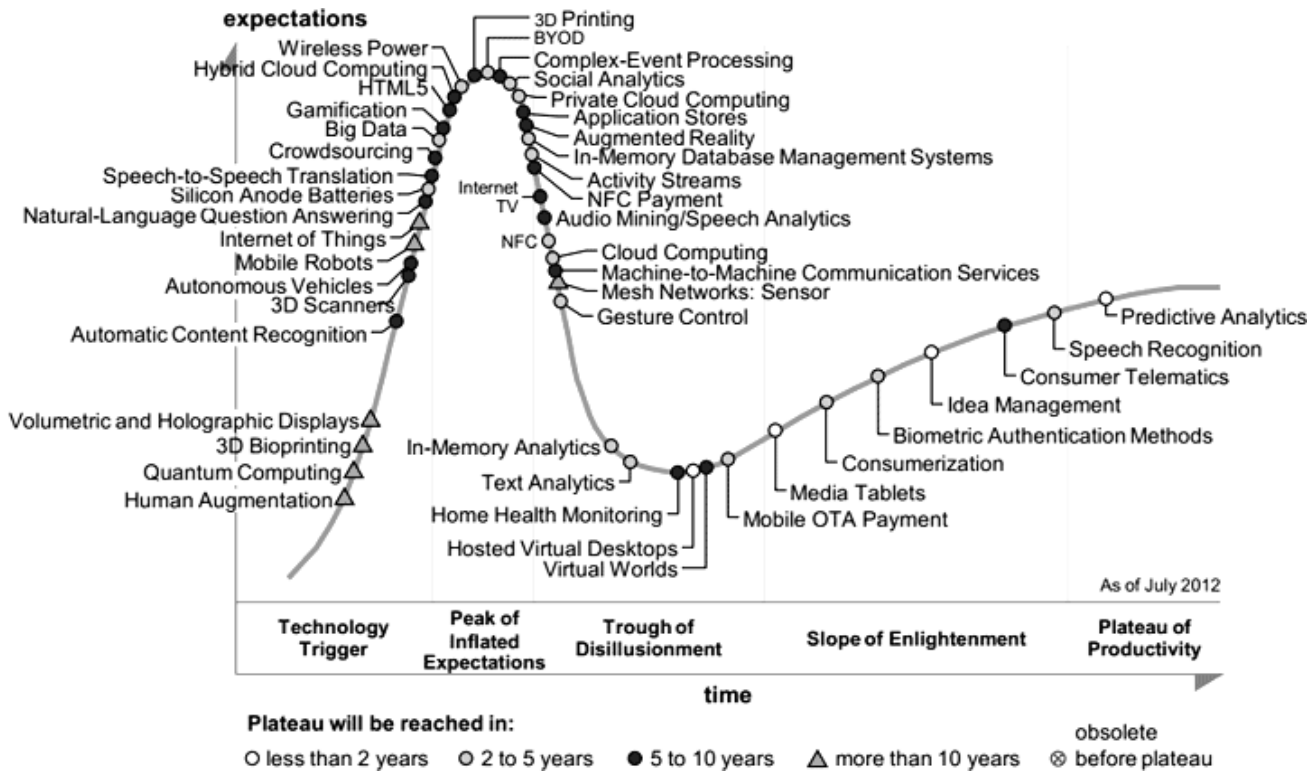


Рис. 1.2. Цикл зрілості технології на 2013 рік за даними Gartner [21]

### Послідовність етапів реалізації технологічної інновації

1. Технологічний тригер (Technology Trigger) – поява інновації, початок публікацій про нову технологію.

2. Пік надмірних очікувань (Peak of Inflated Expectations) – від нової технології очікують революційних властивостей; технологія, завдяки новизні, стає популярною і предметом широкого обговорення у співтоваристві.

3. Позбавлення від ілюзій (Through of Disillusionment) – виявляються недоліки технології, зменшується кількість публікацій, співтовариства фахівців виявляють розчарування новою технологією;

4. Подолання недоліків (Slope of Enlightenment) – усунення основних недоліків, інтерес до технології повільно повертається, технологія починає впроваджуватися у комерційних проектах.

5. Плато продуктивності (Plateau of Productivity) – період зрілості технології, суспільство сприймає технологію як даність, усвідомлюючи її гідності та обмеження.

Не винятком є технології інтелектуального аналізу даних. Відомий у світі експерт у галузі інтелектуального аналізу даних, сховищ даних та CRM, президент

Two Crows Corporation Херб Еделстайн (Herb Edelstein) [22] вважає, що інтелектуальний аналіз даних все ще перебуває на ранньому етапі розвитку, а процес його реалізації на практиці виявляється складнішим, ніж очікувалось. Багато організацій цікавляться цією технологією, але лише деякі активно впроваджують такі проекти. ІТ-команди захопилися міфом про те, що засоби інтелектуального аналізу даних прості у використанні. Вважається, що достатньо застосувати такий інструмент для обробки бази даних, обсягом інформації один терабайт, і миттєво з'явиться корисна інформація. Насправді, успішний ІАД-проект вимагає розуміння суті діяльності, знання даних та інструментів, а також тонкощів процесів аналізу даних. Прикладами використання циклу зрілості сучасних технологій можна навести тривимірний біодрук, мобільні роботи, глобальний перехід бібліотек на електронні носії.

Передбачається, що за допомогою тривимірного біодруку можна буде створювати людські органи, придатні для імплантації. 3D-біопринтер викладає один на одного тонкі шари клітин тканини, керуючись даними докладного сканування друкованого органу. Вже сьогодні на практиці отримують шматки шкіри і кісткові фрагменти, які з успіхом приживляються лабораторним мишам. Фахівці з Gartner [21] вважають, що майбутнє, у якому можна буде просто надрукувати відсутню нирку або навіть серце, настане не швидко. Вже точно, не в найближчі десять років – про це свідчить дуже стримана, якщо не сказати млява, реакція засобів масової інформації на новинку.

Робототехніка існує давно, проте в останні роки публікації про мобільні роботи зустрічаються все частіше і частіше; до речі, їх більше, ніж публікацій про трьохвимірний біодрук. Багато компаній досягають помітних успіхів у розробці роботів-прототипів, здатних переміщуватися – наприклад, американський робот Mable вміє бігати зі швидкістю майже 11 км/год. Правда, йому необхідно для цього "спиратися" на спеціальну стійку, тому бігати він може тільки по колу. До того, як мобільні роботи стануть повсюдно поширені та з'являться у комерційному використанні, пройде ще багато років.

Ідея глобального переходу бібліотек на електронні носії зупинилася на етапі "прізви розчарування". Про об'єднання всіх значних бібліотек світу в одну систему, широко доступну через Інтернет, можна забути на найближче майбутнє. Основною перешкодою для переносу бібліотечних колекцій в Інтернет стало авторське право. Бібліотеки можуть вільно переносити на електронні носії матеріали, що є суспільним надбанням, але в інших випадках необхідно укласти договори з власниками прав. Авторські права на тексти в різних країнах припиняють свою дію через 50-70 років після смерті автора і на Заході ці терміни зростають. Тобто можна говорити лише про локальні успіхи, але немає надії на глобальність проекту.

До кінця поточного десятиріччя буде зростати інтерес до соціальних мереж, хмарних обчислень і мобільної сфери. Приватні хмарні обчислення пішли вгору по кривій очікувань. Серед них є і специфічні, такі як системи управління базами даних в оперативній пам'яті, обробка великих обсягів даних, поглиблена обробка інформації. У довгостроковій перспективі, за межами п'ятирічки, можливі такі застосування: тривимірний друк, контекстно-збагачені сервіси, "віртуальна реальність", Інтернет-ТБ і технології відповіді на природні питання будуть головними технологічними напрямками. Заглядаючи більш ніж на 10 років вперед,

можна з упевненістю сказати, що потенціал мають такі напрями, як 3D біодрук, мобільні роботи і квантові обчислення.

### 1.3. Мережа Байєса – особливий інструмент інтелектуального аналізу даних

Використання мереж Байєса для аналізу процесів різної природи, діяльності людини та функціонування технічних систем дозволяє враховувати та використовувати будь-які вхідні дані – експертні оцінки і статистичну інформацію. Змінні, що використовуються у байєсівських мережах, можуть бути як дискретними, так і неперервними, а характер їх надходження при аналізі та прийнятті рішення може бути і в режимі реального часу, і у вигляді статистичних масивів інформації та баз даних. Завдяки представленню взаємодії між факторами процесу у вигляді причино-наслідкових зв'язків у мережі досягається максимально високий рівень візуалізації та, як наслідок, чітке розуміння суті взаємодії факторів процесу між собою. Саме це відрізняє мережі Байєса від інших методів інтелектуального аналізу даних.

Також перевагами використання мереж Байєса для інтелектуального аналізу даних є можливість врахування невизначеностей статистичного, структурного і параметричного характеру, що особливо важливо актуально для сценарного аналізу, а також для формування висновку за допомогою різних методів – наближених і точних. Загалом можна сказати, що мережа Байєса – це високоресурсний метод ймовірнісного моделювання процесів довільної природи з невизначеностями різних типів, який забезпечує можливість достатньо точного опису їх функціонування, дає змогу оцінювати прогнози та будувати системи управління.

Аналіз існуючих методів інтелектуального аналізу даних показав, що мережі Байєса у порівнянні з популярними моделями “чорних скриньок”, надають зрозуміліше пояснення своїх висновків, припускають логічну інтерпретацію і модифікацію структури відношень між змінними задачі, а також дозволяють у явній формі враховувати попередній апріорний досвід експертів. Завдяки вдалому представленню у вигляді графів, байєсівські мережі дуже зручні для розв'язання прикладних задач користувачів. Вони ґрунтуються на фундаментальних положеннях і результатах теорії ймовірностей, які розроблялися протягом декількох сотень років, що забезпечує їм успіх у розв'язанні практичних задач.

**Байєсівські мережі** – перспективний ймовірнісний інструментарій для моделювання складних ієрархічних процесів (статичних і динамічних) з невизначеностями довільного характеру. Мережа будується у вигляді спрямованого ациклічного графа, який відображає причинні зв'язки між вузлами (змінними) процесу, що досліджується. Редукція спільного розподілу ймовірностей у вигляді добутку умовних ймовірностей, які залежать від малої кількості змінних, дозволяє уникнути “комбінаторних вибухів” при моделюванні.

Взагалі байєсівська методологія насправді набагато ширша, ніж сімейство засобів маніпулювання з умовними ймовірностями в орієнтованих графах. Це і моделі з симетричними зв'язками (випадкові поля та решітки), моделі динамічних процесів (ланцюги Маркова), а також широкий клас моделей із прихованими змінними, що дозволяють вирішувати ймовірнісні задачі класифікації, розпізнавання образів та прогнозування.

**Перспективні області застосування мереж Байєса** такі:

- динамічні процеси і динамічне програмування;
- оптимальне керування стохастичними системами;
- прийняття рішень в автономних інтелектуальних системах.

Байєсівські мережі представляють собою графічні моделі подій і процесів на основі об'єднання деяких результатів теорії ймовірностей і теорії графів. Вони тісно пов'язані з діаграмами впливу, які можна використати для прийняття рішень. Незважаючи на свою назву, ці мережі не обов'язково мають на увазі щільний зв'язок з байєсівськими методами; назва зокрема пов'язана насамперед з байєсівським правилом ймовірнісного висновку [23]. У літературі байєсівські мережі (BN – Bayesian networks) іноді зустрічаються під назвами байєсівські мережі довіри (BBN – Bayesian belief networks), причинні мережі (causal networks) або ймовірнісні мережі (probabilistic networks).

Вони є зручним інструментом для опису досить складних процесів і подій з невизначеностями, особливо корисним при розробці та аналізі машинних алгоритмів навчання. Основною ідеєю побудови графічної моделі мережі Байєса є поняття модульності, тобто розкладання складної системи на прості елементи. Для об'єднання окремих елементів у систему використовуються результати теорії ймовірностей, які забезпечують моделі практичну дієздатність у цілому, а також дають можливість поєднувати графічні моделі з базами даних. Такий граф-теоретичний підхід до побудови моделі дає досліднику можливість будувати моделі процесів з множини сильно взаємодіючих змінних, а також створювати структури даних для наступної розробки ефективних алгоритмів їхньої обробки та прийняття рішень.

Формалізм побудови узагальнених графічних моделей поєднує у собі багато методів статистичного моделювання, таких як факторний аналіз, аналіз розподілів, моделі сумішей розподілів, приховані марковські моделі, фільтри Калмана, моделі Айзінга та деякі інших.

Всі зазначені моделі можна розглянути в рамках графічних моделей байєсівського типу як окремі приклади загального формалізму. Перевагою такого підходу є те, що методи дослідження процесів та обробки даних, розроблені в одній області, можуть бути успішно перенесені в інші.

Незважаючи на те, що байєсівським мережам приділяється багато уваги в закордонній літературі, принципи їхньої побудови, навчання та використання ще недостатньо висвітлені у вітчизняних публікаціях, що істотно ускладнює їх розуміння й застосування.

**Термін “байєсівська мережа”** (Bayesian Network) був запропонований американським вченим Джуді Перлом [24] у 1985 році з метою узагальнення трьох аспектів:

- об'єктивної природи вхідних даних;
- отримання достовірної інформації при застосуванні теореми Байєса;
- ідеї застосування аналізу причин та наслідків, запропонованої в 1763 році у посмертній роботі англійського священника Томаса Байєса [4].

Байєсівські мережі з'явилися на стику двох наук:

- теорії ймовірностей;
- теорії графів (розділ дискретної математики).

Класично ймовірність події означає відношення випадків, коли вона трапилась, до загальної кількості спостережень. Поняття ймовірності відіграє значну роль у повсякденному житті, але математична теорія ймовірностей була сформульована відносно недавно (у XVII столітті). Однак існують декілька *інтерпретацій теорії ймовірності*, серед найбільш домінуючих можна виділити об'єктивістський, суб'єктивістський та логічний [25].

**Об'єктивістський підхід** полягає в тому, що частота появи події при великій кількості спостережень прямує до об'єктивної ймовірності цієї події.

**Суб'єктивістський підхід** розглядає ймовірнісну міру як міру довіри того, як особа ставиться до істинності деякого твердження. Декілька осіб, що мають відношення до події, можуть мати різні ступені довіри до одного й того ж твердження. Термін "байєсівський" часто використовується як синонім суб'єктивної ймовірності.

**Логічний підхід** розглядає ймовірність як ступінь довідності логічно вивіреного висновку. Такий погляд можна розглядати як розширення звичайної логіки.

Ці підходи використовують різні схеми висновку, тобто дві різні школи ймовірнісних розрахунків – школу Паскаля (загальноприйняту) і школу Бекона (індуктивну). У розрахунках за Беконем використовуються правила логіки для доведення або спростування гіпотез, і логічний погляд на ймовірність використовує саме цю школу розрахунків. У розрахунках за Паскалем використовуються байєсівські правила для перевірки і обробки мір довіри, і об'єктивістський та суб'єктивістський підходи використовують цю школу розрахунків.

У 20-му сторіччі індуктивна логіка з одного боку трансформувалася у ймовірнісну логіку, а з іншого – вийшла за межі логіки у власному змісті, знайшовши в істотно збагаченому вигляді нове життя в сучасній математичній статистиці та теорії планування експерименту.

Загалом, еволюція логіки, а згодом і теорії ймовірностей та дискретної математики як науки, відбувалася протягом століть, змінюючись та збагачуючись.

Ще у четвертому столітті до нашої ери Арістотель (384 – 322 рр. до н.е.), на противагу діалектиці Платона (428 – 348 рр. до н.е.), створив формальну логіку. У період раннього середньовіччя (7 – 11 сторіччя) антична логіка сприймалася через призму християнства. Пізніше, у 12-13-му століттях, після того, як всі твори Арістотеля були канонізовані церквою, з'явилася несхоластична логіка.

Перша спроба машинізації процесу логічного висновку була здійснена Раймондом Луллієм (1235 – 1315 рр.). У результаті перших спроб застосування математичного аналізу до визначення вірогідності перемоги у азартних іграх (орлянка, кості, рулетка), як наука, виникла теорія ймовірностей. Леонардо да Вінчі (1452 – 1519 рр.), а пізніше Френсіс Бекон (1561 – 1626 рр.) практично заново винайшли та розвинули ідею індукції та дедуктивного методу [26].

Впродовж XII-XIX ст. відбувалось становлення та розвиток дискретної математики як науки. Значний внесок у розвиток дискретної математики як науки зробили Блез Паскаль (Blaise Pascal, 1623 – 1662 рр.), П'єр Ферма (Pierre de Fermat, 1601 – 1665 рр.), Жозеф-Луї Лагранж (Giuseppe Lodovico Lagrangia; 1736 – 1813 рр.) та Еварист Галуа (Évariste Galois, 1811 – 1831 рр.). Суттєвий вплив на розвиток логіки здійснили праці Рене Декарта (1596 – 1650 рр.), Готфріда Лейбніца (1646 –

1716 pp.) і Джорджа Булля (1815 – 1864 pp.) [26].

Преподобний Томас Байєс (1702–1761 pp.) одним з перших зацікавився ймовірністю настання подій у майбутніх випробуваннях, ґрунтуючись на інформації про минулі випробування. Саме теорема Байєса пов'язує апіорні та апостеріорні ймовірності причин після спостереження за наслідками. У 1763 році у посмертній роботі Т. Байєса “Опис методу розв'язання задачі в рамках доктрини випадкових подій” (“An Essay Towards Solving a Problem in the Doctrine of Chances”), опублікованій його товаришем Річардом Прайсом (Richard Price, 1723 – 1791 pp.) в п'ятдесят третьому номері лондонського журналу Філософського Королівського Товариства [4], було вперше застосовано поняття ймовірності в індуктивному сенсі і встановлені основи ймовірнісного судження та висновку. А практичне застосування теорема Байєса знайшла в теорії ймовірностей одразу після опублікування.

Поняття “байєсівська мережа” було запропоновано Джуді Перлом [8] лише в 1985 році, а до цього часу вони застосовувалися під назвою “каузальні мережі” (causal network), тобто мережі з причинно-наслідковими зв'язками. Ідея впровадження байєсівських мереж полягає у представленні причинно-наслідкових зв'язків процесу у вигляді графа. Назву “байєсівські” мережі отримали завдяки застосуванню в каузальних мережах теореми Байєса. Однак ідея представлення причинно-наслідкових зв'язків моделі у вигляді графів розглядалася ще у 20-х роках ХХ століття. У 1921 році біолог-генетик Сьюел Райт (Sewall Wright, 1889 – 1988 pp.) [27] запропонував пат-аналіз (path analysis) – статистичний метод, що дозволяє оцінити ступінь взаємовпливу змінних у причинно-наслідковій моделі. Для цього він об'єднав лінійні регресійні моделі та спрямовані графи. Пізніше ця ідея була розвинута Гербертом Сімоном (Herbert Simon, 1916 – 2001 pp.) та Хьюбертом Блелоком (Hubert Blalock, 1926-1991 pp.) [28].

Значний внесок у розвиток досліджень у галузі байєсівських мереж зробили Джуді Перл (Judea Pearl), Нір Фрідман (Nir Friedman), Кевін Мерфі (Kevin Murphy), Росс Шахтер (Ross Shachter), Ерік Хорвітц (Eric Horvitz) та Девід Хекерман (David Heckerman). На пост-радянському просторі слід відзначити роботи к.ф.м.н. О. Л. Тулупьєва і його аспірантів С. І. Ніколенка та О. В. Сироткіна, які одними з перших в СНД розпочали дослідження в галузі теоретичних основ байєсівських мереж. Результатом їх спільної роботи стала опублікована у 2006 році монографія “Байесовские сети: логико-вероятностный подход” [7], у якій представлені базові поняття, результати та алгоритми логіко-ймовірнісного висновку в байєсівських мережах довіри. В основу даної монографії покладено положення з книги Джуді Перла “Causality: models, reasoning, and inference” [8]. Також описані алгебраїчні байєсівські мережі, ідеї яких запропоновані В. І. Городецьким [9-11] та розвинені авторами у монографіях [12, 13].

Серед вітчизняних науковців, що зробили значний внесок у розвиток теорії мереж Байєса та її практичного застосуванням, окремо слід відзначити роботи академіка, д.т.н., професора М. З. Згуровського [5], д.т.н., професора Н. Д. Панкратової, д.т.н., професора П. І. Бідюка [5,6], к.т.н. О. М. Терентьєва [5, 6].

**Байєсівські мережі** (БМ) - це графічні моделі для представлення ймовірнісних взаємозв'язків на множині змінних, що поєднують у собі теорію ймовірності та теорію графів. Вони є ефективним інструментом для вирішення проблем невизначеності та опису складних процесів і зокрема відіграють значну

роль у розробці та аналізі алгоритмів машинного навчання. Незважаючи на назву, мережі Байєса необов'язково тісно пов'язані з байєсовою статистикою. Їх назва більше пов'язана з використанням правила Байєса для ймовірнісного висновку [11].

**Формула Байєса** являє собою модель раціонального вибору в умовах неточної або неповної інформації. Процес прийняття рішень може ґрунтуватися на двох протилежних методах – дедукції й індукції. Обираючи шлях дедукції, виходять із певної гіпотези або припущення про сутність події й припускають, що відбудеться згодом, якщо ця гіпотеза достовірна. Оскільки між гіпотезою й передбачуваним розвитком подій існує стійкий зв'язок, дедукція вважається об'єктивним методом, але вона лише підтверджує або спростовує запропоновану гіпотезу й не здатна створити нові знання.

Індуктивні міркування мають протилежну спрямованість: гіпотези будуються й оцінюються на підставі даних досвіду. В основі доказу лежить процес індукції, що відображає перехід від спостережень до закономірностей, що є їх основою. Переваги індуктивного методу аналізу полягають у тому, що за допомогою даних досвіду одержують додаткову інформацію про раніше невідомі явища, будують нові гіпотези й поглиблюють свої знання про зовнішній світ. Однак при цьому неминуче виникає так звана проблема індукції: не можна бути повністю впевненим у правильності доводів. Значно складніше індуктивна диференціальна діагностика, що реалізується у визначенні ймовірності різних визначень, виходячи з наявної інформації й результатів проведених досліджень. Таким чином, дедуктивний метод надійніший і об'єктивніший, але менш продуктивний, ніж індуктивний.

Позначимо через  $\Omega$  *вибірковий простір* подій (або множину подій) випадкових експериментів. Цей вибірковий простір містить всі можливі значення випадкової змінної. За аналогією із словом “стан” слово “змінна” неоднозначне. “Змінна” в імовірнісному контексті дійсно означає змінну, а в граф-теоретичному контексті “змінна” означає атрибут. Якщо змінна приймає конкретне значення, то говорять, що має місце конкретна подія. У теорії оцінювання сигналів, аналізі часових рядів та багатьох інших випадках події називають *спостереженнями*.

Розглянемо дві події  $E \in \Omega$  і  $H \in \Omega$ , що відповідно є спостереженням та гіпотезою. Формула Байєса використовується для обчислення ймовірності того, що подія  $E$  відбудеться за умови, що відбулась подія  $H$ . Тобто для обчислення  $P(E | H)$  - умовної ймовірності  $E$  при заданій події  $H$ .

Умовна ймовірність  $P(E | H)$  дорівнює відношенню сукупної ймовірності подій  $E$  та  $H$   $P(E \cap H)$  до ймовірності події  $H$ , за умови, що вона не дорівнює нулю:

$$P(E | H) = \frac{P(E \cap H)}{P(H)}. \quad (1.1)$$

Аналогічно для події  $H$ :

$$P(H | E) = \frac{P(H \cap E)}{P(E)}. \quad (1.2)$$

Звідси можна **записати правило Байєса** (враховуючи властивість комутативності сукупної ймовірності):



$$P(H | E) = \frac{P(E | H) \cdot P(H)}{P(E)}. \quad (1.3)$$

Цей вираз показує причинно-наслідкові зв'язки між спостереженнями та гіпотезами. Ймовірності  $p(H)$  та  $p(E | H)$  є апіорними, тобто задаються до початку спостережень. Ймовірність  $P(H | E)$  є апостеріорною. Перевага байєсівського методу полягає в тому, що апіорні ймовірності можна уточнювати (оновлювати) у відповідності до реалій протікання процесу, що досліджується. Це дозволяє уточнювати ймовірності подій при надходженні додаткової інформації.

Головне припущення теорії побудови мереж Байєса полягає в тому, що події є вичерпними ( $\cup_{i=1}^n H_i = \Omega$ ) і не перетинаються. При виконанні цих умов ймовірність події  $E$  можна обчислити за допомогою умовних ймовірностей:

$$p(E) = \sum_{i=1}^n p(E \cap H_i) = \sum_{i=1}^n p(E | H_i) \cdot p(H_i). \quad (1.4)$$

Підставивши (1.4) у формулу (1.3), отримуємо вираз:

$$p(H_k | E) = \frac{p(E | H_k) \cdot p(H_k)}{\sum_{i=1}^n p(E | H_i) \cdot p(H_i)}, \quad (1.5)$$

На основі цієї формули будуються мережі Байєса.

У формулі (1.5)  $H_k$  означає будь-яку гіпотезу з  $n$  можливих. Ймовірності  $p(H_k | E)$  задаються експертами *апіорно* або розраховуються за навчальними даними [6, 64]. Тобто їх можна розглядати як відповідь на запитання: "Якою буде ймовірність деякого виміру, якщо відомо, яка гіпотеза була реалізована?". Ймовірності  $p(H_k | E)$  є дуже корисними, тому що, як правило, легше знайти ймовірність послідовності подій типу *причина-наслідок*, ніж навпаки. Значення  $p(H_k)$  називають *апіорними ймовірностями*, вони визначають початкові ймовірності для всіх гіпотез. Сила байєсівського методу полягає в тому, що апіорні ймовірності можна уточнювати (оновлювати) у відповідності до реалій протікання процесу, що досліджується. Це дає можливість уточнювати ймовірності подій при надходженні додаткової інформації. Знаменник виразу (1.5) можна розглядати як нормуючий член, який встановлює значення ймовірності в інтервалі між 0 та 1.

При розв'язанні задач аналізу та моделювання випадкових величин найчастіше використовують нормальний або **Гаусів розподіл**, який має декілька властивостей, що сприяють прискоренню та спрощенню аналізу процесів. Він **описується виразом**:

$$N(x, \mu_x, \sigma_x) = \frac{1}{\sqrt{(2\pi\sigma_x)}} \cdot \exp\left(-\frac{1}{2} \frac{(x - \mu_x)^2}{\sigma_x}\right), \quad (1.6)$$

де  $\mu_x$  - середнє значення процесу;  $\sigma_x$  - стандартне відхилення.

Очевидно, що для опису випадкових процесів можна скористатись будь-яким іншим розподілом, який краще відповідає реальності.

**Іншими типами розподілів** можуть бути такі: бета-розподіл, біноміальний, розподіл  $\chi$  - квадрат, нецентральний розподіл  $\chi$  - квадрат, дискретний рівномірний розподіл, експоненціальний,  $F$  - розподіл, нецентральний  $F$  - розподіл, гамма-розподіл, геометричний, гіпергеометричний, логнормальний, негативний біноміальний та розподіли Пуассона, Рейлі,  $t$  – розподіл Стьюдента, не центрований  $t$  - розподіл, рівномірний неперервний розподіл та розподіл Вейбулла.

**Байєсівська мережа представляє собою** пару  $\langle G, B \rangle$ , у якій перша компонента  $G$  – це спрямований нециклічний граф, що відповідає змінним процесу, що досліджується і записується у вигляді причинно-наслідкової мережі. Друга компонента пари  $B$  – це множина параметрів, що визначають мережу.

Компонента  $B$  містить параметри  $\Theta_{X^{(i)}|pa(X^{(i)})} = P(X^{(i)}|pa(X^{(i)}))$  для кожного можливого значення  $x^{(i)} \in X^{(i)}$  та  $pa(X^{(i)}) \in Pa(X^{(i)})$ , де  $Pa(X^{(i)})$  позначає набір батьків змінної  $X^{(i)} \in G$ .

Кожна змінна  $X^{(i)} \in G$  представляється у вигляді вершини. Якщо розглядають більше одного графа, то для визначення батьків змінної  $X^{(i)}$  в графі  $G$  використовують позначення  $Pa^G(X^{(i)})$ .

**Повна спільна ймовірність мережі Байєса** обчислюється за формулою:

$$P_B(X^{(1)}, \dots, X^{(N)}) = \prod_{i=1}^N P_B(X^{(i)}|Pa(X^{(i)})) \quad (1.7)$$

З математичної точки зору мережа Байєса – це модель подання існуючих і відсутніх імовірнісних залежностей. При цьому зв'язок  $A \rightarrow B$  є причинним, коли подія  $A$  – причина виникнення  $B$ , тобто коли існує механізм, відповідно до якого значення, прийняте  $A$ , впливає на значення, прийняте  $B$ . Мережу Байєса називають причинною (каузальною), коли всі її зв'язки є причинними.

Часто мережі Байєса представляють у вигляді графів із деякими специфічними властивостями. Як і будь-який граф, мережа Байєса складається з вузлів (змінних, вершин), з'єднаних дугами. Дуга, яка з'єднує змінні, вказує на існування причинного зв'язку між ними. Якщо дуга не спрямована, то вона свідчить тільки про наявність зв'язку між змінними. Графи, що містять тільки неспрямовані дуги, називають неспрямованими. Якщо дуга має стрілку, що вказує на напрям залежності – від причини до наслідку, то графи, у яких всі дуги спрямовані, називають спрямованими. А отже, всі байєсівські мережі – спрямовані графи.

**Типи графічних структур байєсівської мережі:**

1. *Дерево* – структура, у якій будь-яка вершина може мати не більше, ніж одну батьківську (рис. 1.3).

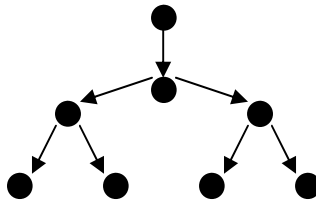


Рис. 1.3. Структура мережі Байєса у вигляді дерева

2. *Полідерево* – структура мережі Байєса, у якій будь-яка вершина може мати більше ніж одну батьківську вершину, але при цьому між будь-якими двома вершинами повинно бути не більше одного з'єднуючого їх шляху (рис. 1.4).

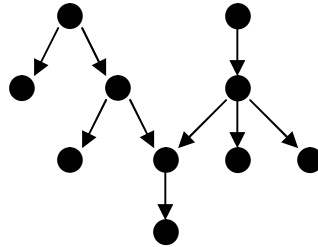


Рис. 1.4. Структура мережі Байєса у вигляді полі-дерева

3. *Мережа* – це структура, у якій будь-яка вершина може мати більш ніж одну батьківську вершину; при цьому між будь-якими двома вершинами може бути більше одного з'єднуючого їх шляху.

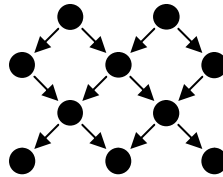


Рис. 1.5. Структура у вигляді мережі

Структури "дерево" та "полідерево" називають також **однозв'язними мережами**, а структури типу "мережа" – **багатозв'язними мережами**.

Як відомо з дискретної математики, якщо всі дуги графа спрямовані, то його називають *спрямованим ациклічним графом*. Якщо спрямований граф містить хоча б один замкнений цикл, то його називають *спрямованим циклічним графом*. Приклади спрямованого ациклічного та циклічного графів показані на рис. 1.6.

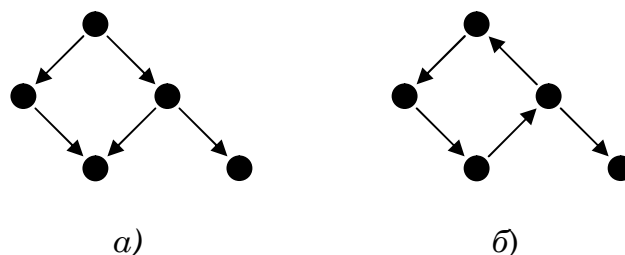


Рис.1.6. Спрямовані ациклічний (а) та циклічний (б) графи

Спрямований ациклічний граф називають *одиначно з'єднаним*, якщо між будь-якими двома його змінними існує тільки один шлях, що їх з'єднує. Графи, показані на рис.1.6, не відносяться до такого типу. Очевидно, що графи з одиначними з'єднаннями (дерева та полі-дерева) є простішими для аналізу, але вони рідко зустрічаються при описанні реальних процесів та подій.

Деякі змінні мереж Байєса мають специфічні назви. Змінну (вузол, вершину) називають *дочірньою* (нащадком), якщо вона залежить від однієї або більше інших змінних. *Батьківська* змінна має одну або більше змінних-нащадків. До множини *нащадків* відносять похідні змінні, які відносяться до однієї батьківської змінної, а

також дочірні змінні дочірніх змінних вищого рівня. Одна і та ж змінна може бути батьківською і дочірньою одночасно. Якщо змінна не має жодної батьківської, то її називають *кореневою*. Кореневі змінні – найважливіші змінні мережі; як правило, це атрибут, який нас цікавить. Змінну називають *листом*, якщо вона не має змінних-нащадків. Однозв'язний граф із змінними, що мають спеціальні назви, представлено на рис. 1.7.

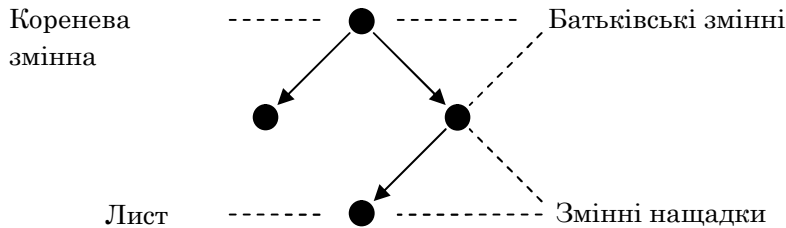


Рис. 1.7. Однозв'язна мережа

*Неоднозв'язність* – це важлива властивість мережі. Якщо мережі однозв'язані, то висновок (рішення) можна знайти досить легко і прямолінійно. Але при вирішенні більшості практичних задач, як правило, мають справу з неоднозв'язними графами (рис. 1.6(б)).

Такі задачі значно ускладнюють і уповільнюють формування висновку.

Розрізняють три типи зв'язків між вершинами:

- лінійний;
- розбіжний (divergent);
- збіжний (convergent).

На рис. 1.8 зображено відповідні типи зв'язків.

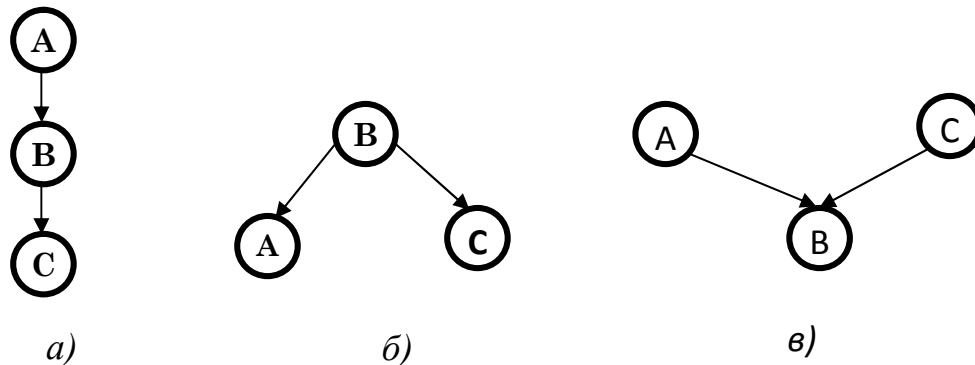


Рис. 1.8. Типи зв'язків

За лінійного типу зв'язку *A* впливає на *B*, а *B*, в свою чергу, на *C* і вершини *A* і *C* є залежними між собою. Але у випадку інстанціювання вершини *B*, ці вершини стають незалежними, бо причина події вже не змінить її наслідків.

При розбіжному типі зв'язку значення вершини *A* може вплинути на значення *C* тому, що інстанціювання *A* змінює значення її батька *B*, яке, в свою чергу, впливає на значення іншого нащадка *C*. Але у випадку інстанціювання вершини *B* нащадки стають незалежними, бо значення вершини *A* вже ніяк не змінить значення вершини *B*.

При збіжному типі зв'язку інстанціювання вершини *A* впливає тільки на значення її нащадка *B*, а вершини *A* і *C* незалежні між собою. Але у випадку

інстанціювання вершини  $B$  значення обох її батьків змінюються, і тепер зміна значення вершини  $A$  вплине на значення вершини  $C$ . Тобто залежність між вершинами з'являється тоді, коли спільний нащадок (або подальші його нащадки) отримує якесь значення.

Узагальнюючи типи зв'язків, можна навести таке означення: дві вершини спрямованого графу  $A$  і  $C$  називаються  $d$ -незалежними, якщо для будь-якого (неспрямованого) шляху з  $A$  в  $C$  існує така проміжна вершина  $B$ , що або зв'язок у цьому шляху лінійний чи розбіжний і вершина  $B$  інстанційована, або зв'язок в шляху є збіжним і вершина  $B$  та жодний з її нащадків не інстанційовані. У протилежному випадку ці вершини будуть називатися  $d$ -залежними [14].

Ключовою властивістю мереж Байєса є те, що графічна  $d$ -незалежність і ймовірнісна незалежність вершин еквівалентні [29].

Визначають такі **типи байєсівських мереж** дискретні, динамічні, неперервні, гібридні.

**Дискретні мережі Байєса** – мережі, у яких змінні вузлів представлені дискретними величинами. **Дискретні мережі Байєса** мають такі **властивості**:

- кожна вершина представляє собою подію, що описується випадковою величиною, яка може мати кілька станів;
- всі вершини, пов'язані з "батьківськими", визначаються таблицею умовних ймовірностей (ТУЙ) або функцією умовних ймовірностей;
- для вершини без "батьків" ймовірності її станів є безумовними (маргінальними).

Інакше кажучи, у байєсівських мережах довіри вершини представляють собою випадкові змінні, а дуги – ймовірнісні залежності, які визначаються через таблиці умовних ймовірностей. Таблиця умовних ймовірностей кожної вершини містить ймовірності станів цієї вершини за умови станів її "батьків". На рис. 1.9 наведено приклад дискретної мережі Байєса, кожна з вершин може приймати один з двох станів  $F$  або  $T$  (скорочення від false та true). Запис  $P(A|B)$  означає ймовірність настання події  $A$  за умови, що подія  $B$  вже відбулась.

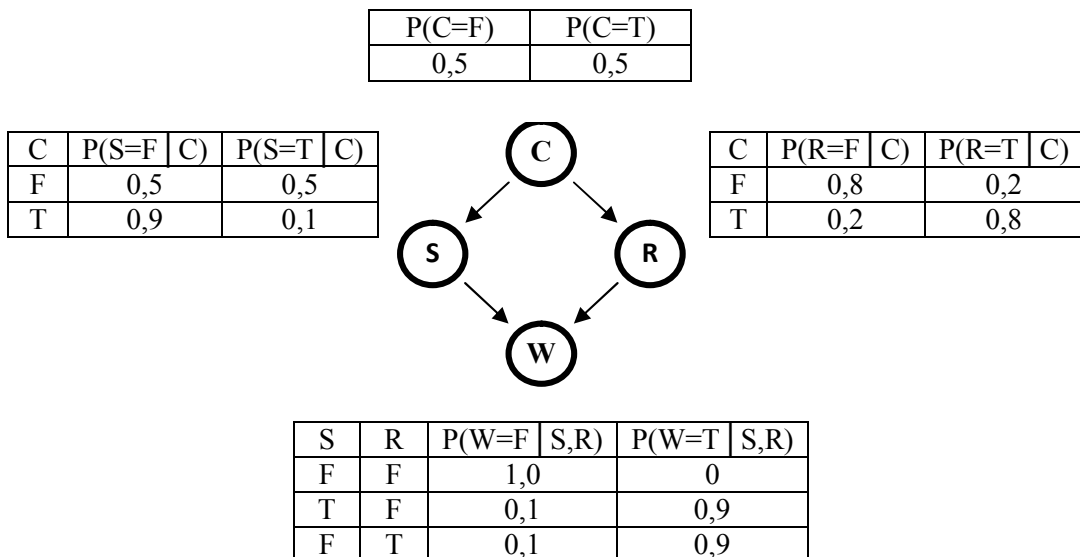


Рис. 1.9. Дискретна мережа Байєса з таблицею умовних ймовірностей

**Динамічні мережі Байєса** – мережі, у яких значення вузлів змінюються з часом. Динамічні мережі Байєса ідеально підходять для моделювання часових процесів. Їх перевага полягає в тому, що вони використовують табличне представлення умовних ймовірностей, що полегшує представлення різних нелінійних явищ [30].

Треба підкреслити, що термін "часова байєсівська мережа" (*temporal Bayesian network*) краще відображає суть, ніж "динамічна байєсівська мережа" (*dynamic Bayesian network*); він передбачає, що структура моделі не змінюється. Зазвичай параметри моделі не змінюються з часом, але до структури мережі завжди можна додати додаткові приховані вузли для опису поточного стану процесу [31].

Найпростіший тип динамічної мережі Байєса – це *прихована модель Маркова* (Hidden Markov Model або скорочено НММ), у якій у кожному шарі наявний один дискретний прихований вузол та один дискретний або неперервний спостережуваний вузол. Ілюстрацію моделі наведено на рис. 1.10. Круглі вершини позначають неперервні вузли, квадратні позначають дискретні;  $X$  – приховані вузли, а  $Y$  – спостережувані. Для визначення динамічної мережі Байєса потрібно задати початковий розподіл  $P(X(t))$ , топологію усередині шару та міжшарову топологію (між двома шарами)  $P(Y(t)|X(t))$  [32].

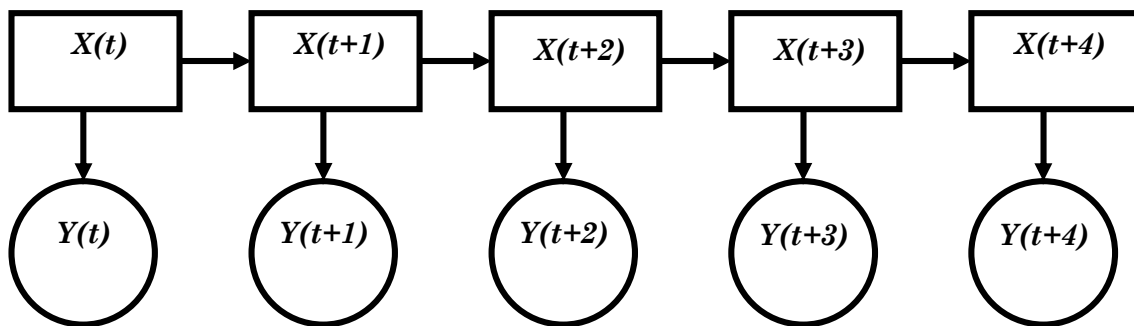


Рис. 1.10. Прихована двохшарова модель Маркова, в якій  $X$  – приховані дискретні, а  $Y$  – дискретні або неперервні спостережувані вузли

У цьому випадку вузли  $Y(t), Y(t+1), Y(t+2), \dots$  представляють собою фонемні слів, а вузли  $X(t), X(t+1), X(t+2), \dots$  – це букви, з яких складається слово. Така модель є динамічною в тому сенсі, щоб представляти собою множину блоків, які повторюються у різні моменти часу [33].

**Неперервні мережі Байєса** – мережі, в яких змінні вузлів – це неперервні величини. У багатьох випадках події можуть приймати будь-які стани з деякого діапазону. Тобто змінна  $X$  буде неперервною випадковою величиною, простором можливих станів якої буде весь діапазон її допустимих значень  $X = \{x | a \leq x \leq b\}$ , який містить нескінченну множину точок. У цьому випадку некоректно говорити про ймовірності окремого стану, тому що при їх нескінченно великій кількості вага кожного буде дорівнювати нулю. Тому розподіл ймовірностей для неперервної випадкової величини визначається інакше, ніж у дискретному випадку; для їх опису використовуються функції розподілу ймовірностей і щільності розподілу ймовірностей.

**Неперервні мережі Байєса** використовують для моделювання стохастичних процесів у просторі станів з неперервним часом.

На рис. 1.11 наведено приклад неперервної мережі Байєса; в цьому випадку використовується розподіл Гауса.

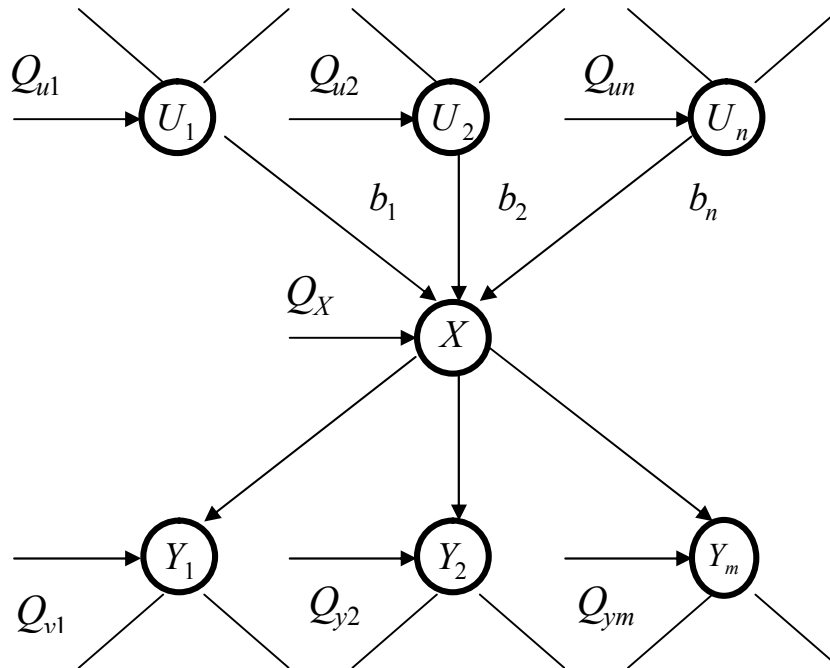


Рис. 1.11. Приклад неперервної мережі Байєса

На рис. 1.11 позначено:  $U_1, U_2, \dots, U_n$  – батьківські вузли для  $X$ ,  $Y_1, Y_2, \dots, Y_m$  – дочірні вузли змінної  $X$ ;  $b_1, b_2, \dots, b_n$  – вагові коефіцієнти,  $Q_x, Q_{u1}, \dots, Q_{un}, Q_{y1}, \dots, Q_{ym}$  – шумові коефіцієнти.

Нехай вузол  $X$  має безліч батьків  $U = \{U_1, U_2, \dots, U_n\}$ , тоді умовний розподіл для  $X$  задається формулою:  $f(X|U_i) = N(x; \mu_x + b_i \cdot \mu_i, \sigma_x)$ , де коефіцієнт  $b_i$  показує зв'язок між  $X$  та його  $i$ -м батьком (його ще називають ваговим коефіцієнтом) [34, 35]:

$$N(x; \mu_x + b_i \cdot \mu_i, \sigma_x) = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_x}} \cdot \exp\left(-\frac{1}{2} \cdot \frac{(x - (\mu_x + b_i \cdot \mu_i))^2}{\sigma_x}\right),$$

де  $\mu$  – математичне очікування,  $\sigma$  – дисперсія, а коефіцієнт  $\frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_x}}$

називають нормуючою константою, яка гарантує, що  $\int_x N(x; \mu_x + b_i \cdot x_i, \sigma_x) = 1$ .

Зв'язок між змінною  $X$  і її батьками ( $U_1, U_2, \dots, U_n$ ) можна представити за допомогою звичайної лінійної регресійної моделі:

$$X = b_1 \cdot U_1 + b_2 \cdot U_2 + \dots + b_n \cdot U_n + Q_x, \tag{1.9}$$

де  $Q_x$  – шумова компонента, що може бути записана у вигляді розподілу Гауса з нульовим математичним очікуванням, а  $b_1, b_2, \dots, b_n$  регресійні коефіцієнти, що показують зв'язок між змінною  $X$  і  $U_1, U_2, \dots, U_n$  – її батьками.

**Гібридні мережі Байєса** – мережі, які містять вузли з дискретними і неперервними змінними. При використанні байєсівських мереж, що містять неперервні і дискретні змінні, існує ряд обмежень:

- 1– дискретні змінні не можуть мати неперервних батьків;
- 2 – неперервні змінні повинні мати нормальний закон розподілу, умовний на значеннях батьків;
- 3 – розподіл неперервної змінної  $X$  з дискретними батьками  $Y$  та неперервними батьками  $Z$  є нормальним розподілом:

$$P(X|Y = y, Z = z) = N(\mu_x(\mu_y, \mu_z), \sqrt{\sigma_x(\sqrt{\sigma_y})}), \quad (1.10)$$

де  $\mu_x, \mu_y, \mu_z$  – математичні очікування,  $\sigma_x, \sigma_y$  – дисперсії,  $\sqrt{\sigma_x}, \sqrt{\sigma_y}$  – середньоквадратичні відхилення;  $\mu_x$  лінійно залежить від неперервних батьків, а  $\sigma_x$  взагалі не залежить від неперервних батьків.

Однак  $\mu_x$  та  $\sigma_x$  залежать від дискретних батьків. Це обмеження гарантує можливість формування точного висновку. Як приклад розглянемо гібридну мережу Байєса, представлену на рис. 1.12.

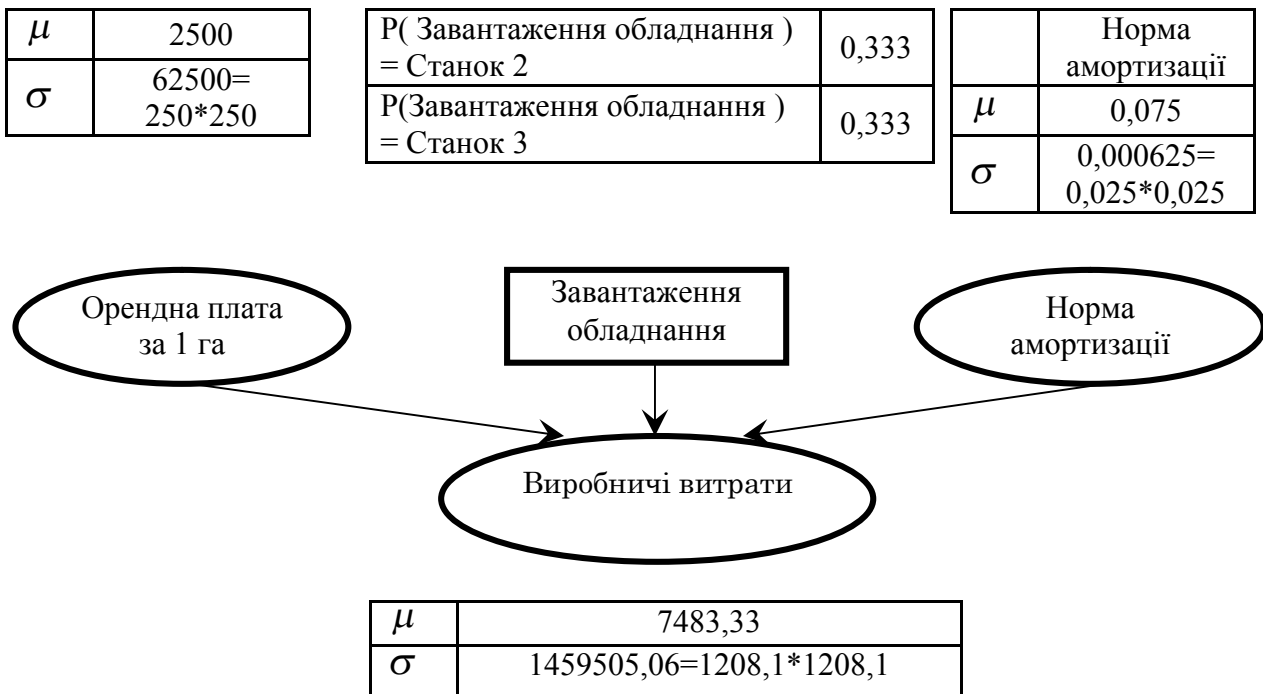


Рис. 1.12. Приклад гібридної мережі Байєса з неперервними та дискретними подіями. Квадратні вершини відповідають дискретним подіям, а овальні – неперервним подіям (гаусівським змінним)



Мережа, представлена на рис. 1.12, дозволяє оцінювати сумарні виробничі витрати залежно від використання та завантаження трьох груп устаткування (наприклад, трьох станків). Відомо, що до складу сумарних виробничих витрат (без врахування зарплати і нарахувань) входять:

- прямі виробничі витрати на кожен групу обладнання за досліджуваний календарний період, які залежать як від кількості використовуваних груп обладнання, так і від часу роботи кожної із груп протягом досліджуваного періоду часу, тобто від ступеня завантаження кожної із груп;
- витрати на амортизацію кожної із груп устаткування, що залежать як від її балансової вартості, так і встановлених норм амортизації;
- орендна плата за ділянку при кожній із груп обладнання, яка використовується для складування сировини та продукції; ця плата залежить як від площі ділянки, так і від ставок орендної плати.

Вершина "Завантаження обладнання" відповідає дискретній події, що характеризується трьома можливими станами. Ймовірність перебування в кожному з них визначається ступенем завантаження кожної із груп устаткування, за умови, що сумарне завантаження всього устаткування дорівнює одиниці. Будемо вважати, що всі групи обладнання завантажені рівномірно. Насправді можливі будь-які інші вихідні розподіли ймовірностей, які враховують різні варіанти завантаження обладнання.

Нехай ставка оренди 1 га землі в середньому становить 2500 у.о. і коливається в межах  $\pm 10\%$ , тобто приймає значення  $2500 \pm 250$  у.о.

Отже, цій вершині відповідають параметри  $\mu = 2500$  та  $\sigma = 62500 = (250)^2$ . А норма амортизації може перебувати в межах 5 - 10 % від балансової вартості, тобто приймати значення  $7,5 \pm 2,5\%$  (або  $0,075 \pm 0,025$  відносних одиниць). Таким чином, цій вершині відповідають параметри  $\mu = 0,075$  та  $\sigma = 0,000625 = (0,025)^2$ .

Що стосується вершини "Виробничі витрати", то вона характеризується випадковою змінною, умовно нормальною на значеннях батьків (тобто на значеннях трьох інших вершин нашого прикладу). Слід зазначити, що в загальному випадку розподіл ймовірностей для вершин, аналогічним вершині "Виробничі витрати", – не просто нормальний, а змішаний нормальний розподіл. Тобто він представляє собою зважену суміш розподілів, для кожного з яких необхідно задати його параметри:

- математичні очікування та дисперсії розподілів, що описують ступінь впливу дискретних батьків;
- вагові коефіцієнти, що враховують ступінь впливу на математичне очікування неперервних батьків.

Припустимо, що в нашому прикладі виконуються такі умови (для трьох станків):

- балансова вартість кожного станка становить 50000, 40000 й 30000 у.о. ;
- площа орендованих ділянок, закріплених за ними дорівнює 0,6; 0,5 та 0,4 га;
- оцінка прямих витрат на підтримку нормальної роботи кожного з станків у середньому становить 3000, 3200 й 3500 у.о. і отримана з 5% точністю.

Ступінь впливу батьківських вершин на "Виробничі витрати" представлено у табл. 1.2.

Таблиця 1.2

Параметри, що визначають розподіл ймовірностей для вершини "Виробничі витрати"

Завантаження обладнання	Станок 1	Станок 2	Станок 3
$\mu$ – математичне очікування	3000	3200	3500
$\sigma$ – середньоквадратичне розсіювання	22 500 = 150*150	25 600 = 160*160	30 625 = 175*175
Норма амортизації	50 000	40 000	30 000
Ставка оренди	0,6	0,5	0,4

Ймовірнісний висновок у таких мережах полягає у поширенні ймовірностей і параметрів гаусівських законів розподілу по всій мережі залежно від отриманих свідчень. В основі процесу формування ймовірнісного висновку лежать досить складні математичні алгоритми. Розглянемо приклад на найпростішій дворівневій мережі для випадку прямого поширення розподілу ймовірностей.

Нехай незалежні дискретні випадкові величини  $X_1, \dots, X_s$  та неперервні випадкові величини  $Z_1, \dots, Z_r$  впливають на результуючу випадкову величину  $Y$ , як показано на рис. 1.13.

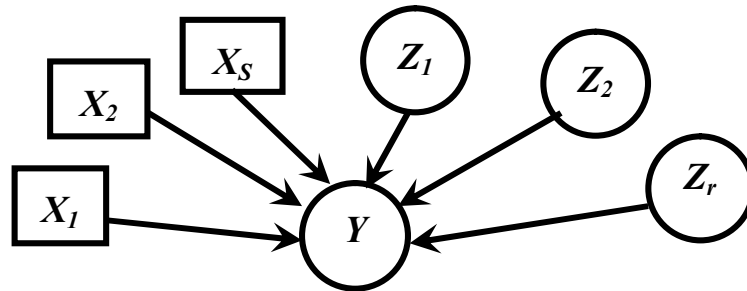


Рис. 1.13. Приклад дворівневої гібридної мережі Байєса з неперервними та дискретними змінними; квадратні вершини відповідають дискретним подіям, а овальні – неперервним подіям

Кожна з дискретних випадкових величин  $X_j (j = 1, \dots, s)$  має своїми результатами значення  $X_{ij} (i = 1, \dots, n_j)$  з ймовірностями  $P_{ij}$ , для яких  $\sum_{i=1}^{n_j} P_{ij} = 1$ .

Спільний вплив дискретних випадкових величин на  $Y$  характеризується математичним очікуванням  $(\mu_{i1}, \dots, \mu_{is})$  та дисперсіями  $(\sigma_{i1}, \dots, \sigma_{is})$ . Кожна з неперервних випадкових величин  $Z_l$  має неперервний нормальний розподіл з параметрами  $(\mu_l, \sigma_l)$ , де  $l = 1, \dots, r$ . Спільний вплив неперервної випадкової величини  $Z_l$  та результатів дискретних величин на результуючу випадкову величину  $Y$  характеризується ваговими коефіцієнтами  $k_{l,i1, \dots, is}$  для  $l = 1, \dots, r$ .

Тоді характеристики результуючої величини  $Y$  можуть бути обчислені за формулами 1.11 та 1.12:

$$\mu = \sum_{i1}^{n1} \dots \sum_{is}^{ns} p_{1,i1} \dots p_{s,is} \left( \mu_{i1, \dots, is} + \sum_{l=1}^r K_{l,i1, \dots, is} \cdot \mu_l \right), \quad (1.11)$$

$$\sigma = \sum_{i1}^{n1} \dots \sum_{is}^{ns} p_{1,i1} \dots p_{s,is} \cdot \left( \left( \mu_{i1, \dots, is} + \sum_{l=1}^r K_{l,i1, \dots, is} \right)^2 + \sigma_{i1, \dots, is} + \sum_{l=1}^r K_{l,i1, \dots, is}^2 \cdot \sigma_l \right) - \mu^2 \quad (1.12)$$

Стосовно розглянутого прикладу, який містить дві вихідні неперервні ( $r=2$ ) змінні та одну дискретну ( $s=1$ ) змінну, що має три результати ( $n_1=3$ ), числові характеристики випадкової змінної "Виробничі витрати" складуть:

$$\begin{aligned} \mu &= \sum_{i=1}^3 p_i \left( \mu_i + \sum_{l=1}^2 k_{li} \cdot \mu_l \right) = \\ &= 0,333 \cdot (3000 + 50000 \cdot 0,075 + 0,6 \cdot 2500) + \\ &+ 0,333 \cdot (3200 + 40000 \cdot 0,075 + 0,5 \cdot 2500) + \\ &+ 0,333 \cdot (3500 + 3000 \cdot 0,075 + 0,4 \cdot 2500) = \\ &= 0,333 \cdot (8250 + 7450 + 6750) = 7483,33 \\ \sigma &= \sum_{i=1}^3 p_i \left( \left( \mu_i + \sum_{l=1}^2 k_{li} \cdot \mu_l \right)^2 + \sigma_i + \sum_{l=1}^2 k_{li}^2 \cdot \sigma_l \right) - \mu^2 = 1459505,6, \\ \sqrt{\sigma} &= 1208,1. \end{aligned}$$

### Контрольні питання

- 1 Технологія інтелектуального аналізу даних: наведіть означення та приклади практичного застосування.
- 2 Що означає у даному випадку слово «інтелектуальний»?
- 3 Дайте пояснення інтелектуальному аналізу даних як мультидисциплінарній області знань. У чому полягають особливості розвитку цієї області?
- 4 Наведіть означення та приклад постановки і розв'язання задачі машинного навчання.
- 5 Наведіть означення та приклад розв'язання задачі методами штучного інтелекту.
- 6 Дайте означення життєвого циклу технології, наведіть графік та приклад.
- 7 Наведіть приклади популярних задач і методів інтелектуального аналізу даних.
- 8 Наведіть приклади алгоритмів, які використовуються в інтелектуальному аналізі даних.
- 9 Історія виникнення мереж Байєса. Ким був Томас Байєс і в якому столітті він жив?
- 10 Наведіть приклади сфер людської діяльності, у яких успішно

використовуються мережі Байєса?

- 11 Яка головна мета побудови мережі Байєса?
- 12 Які типи невизначеностей враховуються у мережах Байєса?
- 13 Які два типи навчання необхідно виконати для побудови мереж Байєса?
- 14 Чим відрізняються дискретні змінні від неперервних?
- 15 Чи може мережа Байєса містити одночасно дискретні і неперервні змінні?
- 16 Наведіть недоліки і переваги використання мереж Байєса?
- 17 Які методи інтелектуального аналізу даних близькі за своїм призначенням до байєсівських мереж?

## РОЗДІЛ 2 РОЗВ'ЯЗАННЯ СЛАБКОСТРУКТУРОВАНИХ ЗАДАЧ СИСТЕМНОГО АНАЛІЗУ ЗА ДОПОМОГОЮ БАЙЄСІВСЬКИХ МЕРЕЖ

- Застосування мереж Байєса до розв'язання слабкоструктурованих задач
- Огляд випадків навчання мереж Байєса
- Алгоритми навчання мереж Байєса з прихованими вершинами
- Побудова структури мережі Байєса з прихованими вершинами та невідомою структурою
- Методика знаходження параметрів байєсівських мереж з прихованими вершинами
- Приклади застосування мереж Байєса з прихованими вершинами

### 2.1. Застосування мереж Байєса до розв'язання задач системного аналізу

Сучасні тенденції соціально-економічного розвитку, суспільного прогресу і політичних процесів потребують розв'язання багатьох нагальних завдань. Однією із головних проблем, яка пов'язана з виконанням цих завдань, є те, що дуже часто задачі носять слабкоструктурований характер. Такі проблеми виникають у складних технологічних, фінансово-економічних, соціальних, екологічних і біологічних системах. Вони характеризуються умовами невизначеності різної природи, наявністю неповної, нечіткої інформації та ускладнюються впливом людського фактору. Задачі такого типу розв'язуються методами системного аналізу і теорії прийняття рішень.

**Системний підхід** застосовують у наукових дослідженнях і при розв'язанні практичних задач, пов'язаних з прогнозуванням, проектуванням і управлінням у технічних (технологічних) системах, біології, психології, соціально-економічній, політичній і військовій сферах. Починаючи з кінця 50-х років минулого століття, ці методи застосовують при прийнятті управлінських рішень у промисловості, фінансовій, комерційній діяльності та інших областях. Системний підхід передбачає виявлення і дослідження зв'язків та відношень між елементами об'єктів управління, встановлення ієрархічності структури та аналіз і врахування невизначеностей з метою підвищення якості управлінських рішень.

**Рішенням** вважається обґрунтована множина дій з боку особи, що приймає рішення (ОПР), спрямованих на об'єкт чи систему управління, який надає можливість привести даний об'єкт чи систему до бажаного стану або досягнути поставленої мети. Рішення є одним із видів розумової діяльності і проявом волі людини.

**Характерними ознаками рішення** є такі:

- можливість вибору з набору альтернативних варіантів: за відсутності альтернатив відсутній і вибір, отже, відсутнє й рішення;
- наявність мети: безцільний вибір не розглядається як рішення;
- необхідність вольового акту особи, що приймає рішення при виборі альтернативи, тому що вона формує рішення при боротьбі мотивів і думок.

**Прийняття рішення** – це процес вибору найбільш преференційного рішення з множини допустимих рішень або упорядкування множини рішень. Прийняття рішень можливе на підставі знань про об'єкт управління, процеси, що в ньому відбуваються і можуть відбутися з перебігом часу, а також за наявності множини показників, що характеризують ефективність та якість прийнятого рішення. Тобто необхідні адекватна модель об'єкту і модель прийняття та оцінювання прийнятого рішення. Під моделлю прийняття рішень мається на увазі формальне подання поставленої задачі та процесу прийняття рішень.

Переважає більшість досліджень виконується за припущення, що дані є повними, тобто значення (виміри) усіх змінних відомі для кожного варіанту моделювання. Однак це не зовсім реальні ситуації, адже приховані змінні відіграють ключову роль при розв'язанні багатьох практичних задач. Невідомий регулюючий механізм може бути ключем до підвищення якості аналізу біологічних та систем іншої природи. Зв'язки між симптомами та іншими явищами можуть дати поштовх до розгадки прихованої фундаментальної проблеми при діагностуванні та управлінні складними системами. Навмисно або ненавмисно замаскована економічна сила може бути причиною виникнення взаємно пов'язаних фінансових феноменів, які відіграють вирішальну роль для розвитку процесу [36]. Приховані змінні можуть відігравати роль деякого підсумкового механізму, що "перехоплює" інформацію із спостережуваних змінних та "спрямовує" її до іншої частини мережі. Тому введення і використання прихованих змінних при моделюванні може надати можливість значно спростити структуру моделі (байєсівської мережі) і в результаті отримати високоякісний загальний результат, тобто ймовірнісний висновок.

**Байєсівські мережі з прихованими змінними** – це чіткий приклад застосування системного підходу до розв'язання слабкоструктурованих задач. Знаходження причинно-наслідкових зв'язків, обробка неповних даних, отримання адекватної ймовірнісної моделі на виході – все це робить мережі Байєса ефективним інструментом системного аналізу. А головна потужність та цінність мереж Байєса у рамках системного підходу полягає у їх здатності виявляти неочевидні, нетривіальні та невідомі зв'язки між факторами, про які іноді самі експерти у відповідній предметній області не завжди можуть мати достатню уяву. Саме тому вони набули широкого застосування в найрізноманітніших сферах сучасного життя людини і продовжують активно досліджуватись та впроваджуватись у практику.

**Системний аналіз** – це науковий метод пізнання, який ґрунтується на визначеній послідовності дій, спрямованих на виявлення, обґрунтування та використання структурних зв'язків між змінними та/або елементами досліджуваної системи. Він спирається на комплекс загальнонаукових теоретичних, експериментальних, природничих, математичних, статистичних та інших методів дослідження. Системний аналіз має універсальний міждисциплінарний характер.

Одне з відомих означень системного аналізу характеризує його як фундаментальну теоретичну основу, яка може використовуватись для подолання складностей і труднощів дослідження об'єктів та процесів у самих різних галузях людської діяльності з подальшою метою створення моделей для прогнозування, передбачення та прийняття коректних управлінських рішень. Тому для розв'язання практичних задач розглянемо поняття саме прикладного системного аналізу.

**Прикладний системний аналіз** – це наукова дисципліна, яка на основі об'єктивно вибраних та системно організованих, структурно взаємопов'язаних та

функціонально взаємодіючих формалізованих і евристичних процедур, методологічних та методичних засобів, апарату математико-статистичного аналізу, програмного забезпечення і комп'ютерних систем забезпечує отримання і накопичення інформації стосовно функціонування об'єкта дослідження в умовах наявності невизначеностей структурного, параметричного і статистичного характеру з подальшою метою формування знань, достатніх для розв'язання поставленої задачі (прогнозування, передбачення та управління).

Більшість рішень у сучасних складних задачах приймаються людьми одноосібно або колегіально в умовах наявності невизначеностей різної природи та типів. **Невизначеність** припускає наявність факторів, при яких результати дій не є детермінованими, а ступінь можливого впливу цих факторів на результати невідома.

Аналіз умов появи невизначеностей може виконуватись на абстрактному теоретичному рівні або з точки зору конкретної ситуації, наприклад, з точки зору виявлення можливості побудови математичної моделі чи з точки зору теорії інформації, де невизначеність виступає як характеристика ситуації вибору. Категорія невизначеності характеризується деякими змінними параметрами, які описують різні види невизначеностей: глобальну невизначеність, ситуативну, політичну, соціальну і т. ін. При розв'язанні задач прийняття рішень в умовах наявності невизначеностей необхідно встановити рівень аналізу і типи невизначеностей, що розглядаються.

Необхідно зазначити, що часто невизначеність ототожнюють лише з відсутністю повної інформації про той чи інший об'єкт. Насправді недостатні знання станів об'єкту – це не єдина невизначеність. Поряд з цим іноді можна розглядати невизначеність цілей та невизначеність критеріїв вибору рішень. У багатьох реальних задачах складність прийняття рішення визначається насамперед кількістю альтернативних варіантів та кількістю і різномірністю критеріїв оцінювання цих варіантів.

Загалом **задачі прийняття рішень розділяють** на три групи: структуровані, слабко структуровані та неструктуровані. Виділення добре структурованих, слабкоструктурованих і неструктурованих задач звичайно досить умовне, оскільки досягти чіткого розмежування тут неможливо. **Добре структуровані завдання** упорядковані і їх розв'язання – це процеси, що повторюються; крім того, вони однозначні, оскільки кожне завдання має єдиний (кращий) метод розв'язання. Для розв'язання **слабкоструктурованих завдань** існує більше можливих методів, і самі рішення можуть досить суттєво відрізнятися одне від одного. **Неструктуроване завдання** або взагалі не має відомих методів рішення, або може мати їх надто багато.

Розглядаючи докладніше **слабкоструктуровані проблеми**, можна виділити їх такі **особливості**:

- задача не може бути виражена у числовій формі;
- цілі не можуть бути вираженими у термінах точно визначеної цільової функції;
- практично не існує алгоритмічного розв'язку задачі;
- якщо алгоритмічний розв'язок існує, то його не можна використати через обмеженість ресурсів (час, пам'ять і таке інше).

До слабкоструктурованих проблем відносять найбільш складні завдання стратегічного планування і розподілу ресурсів, фінансово-економічні та задачі

технічного, політичного, військово-стратегічного характеру. Розв'язання проблем, що мають "слабкоструктурований характер", і є основним завданням системного аналізу.

Слабкоструктуровані проблеми, як правило, пов'язані з розробкою стратегічних довгострокових курсів діяльності, кожен з яких охоплює багато аспектів діяльності організації і реалізується поетапно. Наприклад, визначення стратегії випуску нової продукції, технічного переозброєння виробництва, удосконалення організації управління і таке інше. Ці проблеми містять разом з добре вивченими елементами, що кількісно формалізуються, також невідомі або невимірювані компоненти, які відрізняються значною невизначеністю. Вони вирішуються за допомогою методів імітаційного математичного моделювання і системного аналізу, які поєднують у собі складні математичні розрахунки з великим об'ємом суб'єктивних суджень керівників і фахівців.

Для розв'язання слабкоструктурованих задач використовують методологію системного аналізу, яка дає можливість проектувати, реалізовувати і використовувати на практиці сучасні інформаційні системи підтримки прийняття рішень (СППР).

Необхідно відзначити, що моделювання – це один з найбільш універсальних способів вивчення різноманітних процесів та явищ. У наукових дослідженнях та інженерній практиці широко використовують чисельні методи та прийоми моделювання. При цьому розрізняють фізичне і математичне моделювання.

При фізичному моделюванні створена модель відображає поведінку об'єкта, що вивчається, із збереженням його фізичної природи та принципів функціонування. Між об'єктом, що вивчається, та його моделлю мають бути збережені деякі співвідношення подібності, що впливають із закономірностей фізичної природи явища і забезпечують можливість використання даних, які отримують за допомогою моделювання з метою оцінювання властивостей і характеристик досліджуваного об'єкта. Фізичне моделювання має досить обмежену сферу застосування внаслідок значних матеріальних та технічних труднощів, які можуть при цьому виникати.

Набагато ширші можливості має математичне моделювання.

**Математичне моделювання** – це спосіб дослідження об'єктів на основі вивчення явищ, що мають різний фізичний зміст, але можуть описуватись однаковими (подібними) математичними співвідношеннями. При цьому співвідношення подібності повинні зберігатись між змінними математичної моделі та найбільш важливими властивостями і характеристиками об'єкта, що вивчається.

На практиці часто використовують багато різновидів математичних моделей, що засновані на поєднанні можливостей сучасної математики, прикладної статистики і обчислювальної техніки, наприклад, графічні та імітаційні моделі. Так, графічна модель оперує системою взаємопов'язаних креслень і зображень для відображення реальних взаємозалежностей, характерних для досліджуваного об'єкта. Імітаційна модель представляє собою систему взаємопов'язаних комп'ютерних програм для імітації поведінки об'єкта.

Як сказано вище, **об'єктом системного аналізу є слабкоструктуровані проблеми**. Тому його методи спрямовуються на "зняття" цієї слабкої структурованості об'єктів шляхом структуризації (упорядкування) задачі, тобто вибору ієрархії цілей, завдань, установлення взаємних зв'язків між елементарними



об'єктами, що визначають масштабність системи, керуючих та інформаційних зв'язків, ієрархію показників ефективності та їх динаміку.

У своєму розвитку системний аналіз традиційно спирався на методи дослідження операцій, які у своїй більшості використовують математичний апарат, який забезпечує отримання високоякісних рішень. Долаючи цю однобічність методології дослідження операцій, системний аналіз вводить у свою методологію додаткову логічну складову для поглиблення дослідження проблем управління. І, залежно від проблеми, системний аналітик приймає рішення стосовно використання тих або інших методів.

Узагальнена методика вирішення неструктурованих і слабкоструктурованих багатокритеріальних задач ґрунтується на створенні інформаційних систем підтримки прийняття рішень.

Ще одним науковим напрямом, що лежить в основі дослідження слабкоструктурованих систем, є методологія когнітивного моделювання. У рамках когнітивної моделі інформація про систему представляється у вигляді набору понять (факторів) і пов'язуючої їх причинно-наслідкової мережі, що має назву *когнітивна карта*. Така карта є віддзеркаленням суб'єктивних представлень експерта (чи групи експертів) про закони і закономірності, властиві модельованій системі. До когнітивної карти застосовують методи аналітичної обробки, орієнтовані на дослідження структури системи і отримання прогнозів її поведінки при різних управляючих подіях з метою синтезу ефективних стратегій управління.

Для розв'язання складних, у тому числі слабкоструктурованих, проблем розроблено **процедуру прийняття рішень, що складається з таких етапів:**

- 1) формулювання проблемної ситуації;
- 2) визначення цілей;
- 3) визначення критеріїв досягнення цілей;
- 4) побудова моделей для обґрунтування рішень;
- 5) пошук оптимального (допустимого) варіанту рішення;
- 6) узгодження рішення;
- 7) підготовка рішення до реалізації;
- 8) затвердження рішення;
- 9) управління ходом реалізації рішення;
- 10) перевірка ефективності рішення.

Основними інструментами розв'язання складних проблем є:

- імітаційне моделювання;
- кореляційний, регресійний та кластерний аналіз;
- методи статистичної класифікації;
- методи аналізу ієрархій;
- теорія матриць;
- теорія експертних оцінок;
- нечітка логіка;
- нейронні мережі;
- генетичні алгоритми;
- мережі Байєса.

Отже, в багатьох ситуаціях прийняття рішень щодо слабкоструктурованих проблем, коли використовуються переважно експертні знання про предметну

область, основу яких становлять набори правил та результати суб'єктивного аналізу ситуації, доцільно використовувати саме ймовірнісні інструменти, які дозволяють розкривати невизначеності шляхом обчислення ймовірностей станів вершин мережі Байєса на основі наявної інформації про значення інших вершин, що дає змогу визначити оптимальний варіант переходу системи на наступного стану.

Тому сьогодні мережі Байєса все частіше використовуються у системах підтримки прийняття рішень, призначених для вирішення слабо структурованих задач в умовах невизначеності, коли необхідно врахувати статистичні та структурні невизначеності процесів різної природи.

## 2.2. Огляд випадків навчання мереж Байєса

Для опису байєсівської мережі необхідно визначити топологію графа й параметри кожного вузла. Цю інформацію можна одержати з навчальних даних, але одержання правильної топології мережі є більш складним завданням, ніж одержання параметрів вершин.

Особливий підхід необхідно застосовувати у випадку, коли деякі вершини приховані або коли маємо справу з некоректними чи неповними даними.

Виділяють чотири випадки навчання мережі Байєса, опис яких наведено в таблиці 2.1.

Таблиця 2.1

Чотири випадки навчання мережі Байєса

Структура	Спостереження	Метод
Відома	Повне	Оцінка максимальної правдоподібності
Відома	Часткове	Максимізація математичного сподівання або пошук екстремуму
Не відома	Повне	Пошук у просторі моделей
Не відома	Часткове	Структурний алгоритм максимізації математичного сподівання

Розглянемо випадок *навчання, коли відома структура, спостереження повні*, тобто *немає прихованих вершин*.

У цьому випадку обчислюються значення параметрів кожного умовного імовірнісного розподілу, які максимізують правдоподібність навчальних даних. Нормалізоване логарифмічне рівняння правдоподібності для навчальної множини  $D = \{D_1, \dots, D_m\}$  має вигляд:

$$L = \frac{1}{N} \cdot \sum_{i=1}^n \sum_{m=1}^M \log( P(X_i | Pa(X_i), D_m) ), \quad (2.1)$$

де  $Pa(X_i)$  означає набір батьків  $X_i$ ,  $D$  – навчальна вибірка, а  $D_m$  – це 1-й елемент навчальної множини,  $M$  – кількість подій у навчальній множині.

**Приклад.** Дано просту мережу для фіктивного медичного прикладу, що описує наявність у пацієнта метастатичного раку мозку на основі головних болів, коми та наявності кальцію у сироватці крові [110]. Схема мережі зображена на рис. 2.1.

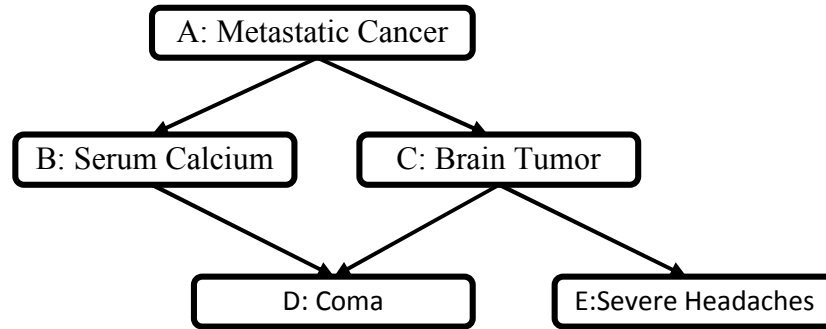


Рис. 2.1. Байєсівська мережа для задачі "Рак мозку" (випадок навчання, коли немає прихованих вершин)

Кожна вершина має по 2 стани: "наявний" і "відсутній" для вершин Metastatic Cancer, Coma, Brain Tumor, Severe Headaches; "підвищений" та "в нормі" для вершини Serum Calcium.

Нехай є дані для навчання, тобто інформація про те, скільки разів була у пацієнтів кома, а причиною цього було те, що були рак мозку і підвищений кальцій у крові  $N(D=1, C=1, B=1)$ , скільки разів спостерігалась кома і був підвищений кальцій в крові, але не було пухлини в мозку  $N(D=1, C=0, B=1)$  и так далі. Використовуючи ці дані, оцінка максимальної правдоподібності вершини  $D$  обчислюється як:

$$\begin{aligned}
 P(D = d \mid B = b, C = c) &= \frac{N(D = d, C = c, B = b)}{N(C = c, B = b)} = \\
 &= \frac{N(D = d, C = c, B = b)}{N(D = 0, C = c, B = b) + N(D = 1, C = c, B = b)}
 \end{aligned}$$

Якщо вершини описані функцією Гауса, то можна підрахувати вибіркове середнє і дисперсію, а потім за допомогою лінійної регресії обчислити матрицю ваг. Для інших видів розподілу використовують більш складні процедури.

**У випадку невідомої структури та повних спостережень** найбільш ймовірною моделлю в даному випадку є повний граф, тому що в цьому випадку буде задіяно найбільшу кількість параметрів; отже, така модель найбільше відповідатиме даним.

Формула Байєса (формула 1.1.) має вигляд:

$$P(G \mid D) = \frac{P(D \mid G) \cdot P(G)}{P(D)},$$

де  $G$  - спрямований ациклічний граф, що відповідає випадковим змінним, а  $D = \{x^1, \dots, x^N\}$  - множина даних.

Прологарифмуємо цю формулу:

$$\log(P(G \mid D)) = \log(P(D \mid G)) + \log(P(G)) + (-\log(P(D))), \quad (2.2)$$

У формулі (2.2) доданок  $-\log(P(D))$  грає роль штрафуючої компоненти за надмірну складність моделі. Складова  $P(D \mid G)$  також може виступати штрафуючою компонентою за надмірно складні моделі (цей випадок відомий як лезо Оккама).

Для виконання точних розрахунків, пов'язаних з вибором моделі, необхідно обчислити  $P(D) = \sum_G P(D | G)$ , що є завданням експоненційної складності. Замість цього можна використовувати байєсівський інформаційний критерій, який визначається у даному випадку як:

$$\log(P(G | D)) \approx \log(P(D | G, \hat{\theta}_G)) - \frac{\log(N)}{2} \cdot \dim(G), \quad (2.3)$$

де  $N$  - кількість моделей;  $\dim(G)$  - розмірність моделі (кількість вільних параметрів);  $\hat{\theta}_G$  - максимально правдоподібна оцінка параметрів,  $-\frac{\log(N)}{2} \dim(G)$  - штрафуюча компонента за надмірно складні моделі.

Наступним кроком після вибору структури є навчання структури таким чином, щоб направлений ациклічний граф найкраще задовольняв даним. Це завдання є NP-складною задачею, тобто задачею з нелінійною поліноміальною оцінкою числа ітерацій. Тому зазвичай використовують локальні алгоритми пошуку, такі, як наприклад, жадібний метод пошуку екстремуму (greedy hill climbing method) або метод гілок і границь для пошуку в просторі графів.

Загалом, всі існуючі сучасні методи побудови структури мереж Байєса, як відзначають Дж. Ченг та В. Лиу [37, 38] можна умовно розділити на дві категорії:

- на основі оціночних функцій (search & scoring);
- методи із застосуванням тестів на умовну незалежність (dependency analysis).

**Навчання мереж Байєса з прихованими вершинами** має свої особливості. Багато реальних задач характеризується наявністю прихованих змінних (так званих латентних змінних), які не є спостережуваними і доступними для навчання [39].

Наприклад, історії хвороби часто включають описи спостережуваних симптомів, лікування, що застосовується, і, можливо, результати лікування, але рідко містять відомості про безпосереднє спостереження за розвитком самого захворювання. Очевидне питання: "Якщо хід розвитку захворювання не спостерігається, то чому б не побудувати модель без нього"? Відповідь на це питання можна знайти на рис. 2.2, де показана невелика фіктивна діагностична модель для серцевого захворювання. Існують три спостережувані фактори, що сприяють або перешкоджають розвитку захворювання, і три спостережувані симптоми. Передбачається, що кожна змінна має три можливі стани (наприклад, none (відсутній), moderate (помірний) і severe (серйозний)). Видалення прихованої змінної з мережі, наведеної на рис. 2.2a), спричиняє створення мережі, показаної на рис. 2.2b), і при цьому загальна кількість параметрів збільшується з 78 до 708. Таким чином, приховані змінні дозволяють різко зменшити кількість параметрів, потрібних для визначення байєсової мережі.

А даний факт, у свою чергу, дозволяє різко зменшити об'єм даних, необхідних для визначення в процесі навчання даних параметрів.

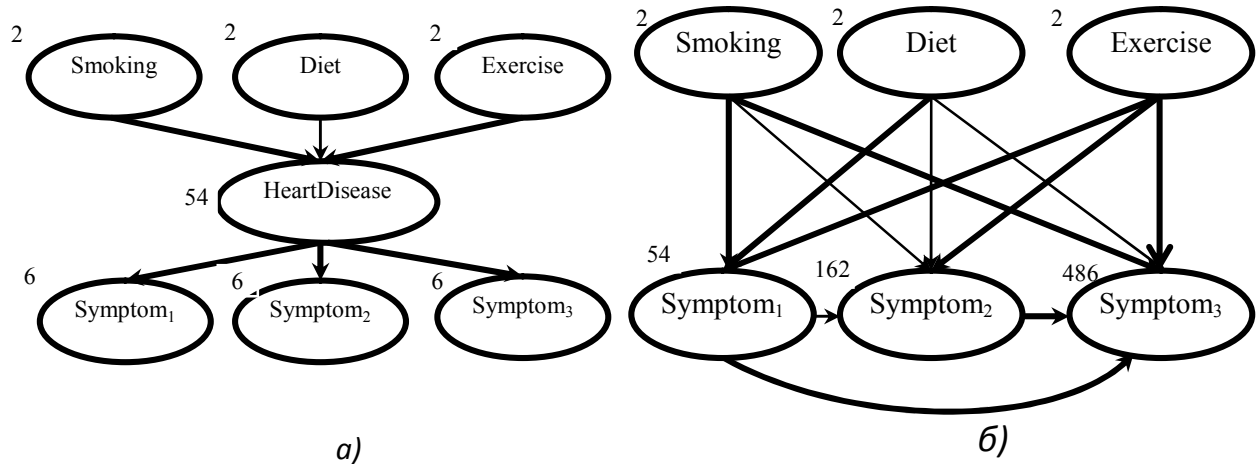


Рис. 2.2. Ілюстрація використання прихованих змінних

Приховані змінні важливі, але їх введення призводить до ускладнення задачі навчання. Наприклад, на рис. 2.2а не показано, як знайти в процесі навчання розподіл умовних ймовірностей для змінної HeartDisease, якщо задані її батьківські змінні, оскільки значення змінної HeartDisease у кожному випадку невідоме; така ж проблема виникає при пошуку в процесі навчання розподілів ймовірностей для симптомів.

Таким чином, перевагами використання прихованих вершин у мережах Байєса є:

- значне спрощення структури мережі;
- зменшення кількості параметрів, необхідних для визначення мережі Байєса, а це в свою чергу дозволяє різко зменшити об'єм даних, необхідних для визначення у процесі навчання цих параметрів.

Недоліки використання прихованих вершин в мережах Байєса:

- ускладнення задачі навчання мережі Байєса;
- у разі невідомої структури мережі невирішеною залишається проблема визначення кількості прихованих вершин, місце їх розташування та кількість станів кожної з них.

Іноді інформація щодо прихованих вершин в мережах Байєса є. Проте в загальному випадку не має ніяких даних щодо прихованої вершини. Відповідно постає дві основні проблеми навчання мереж Байєса з прихованими вершинами:

1. Скільки станів повинна мати прихована вершина?
2. Як умовні ймовірності можуть бути знайдені без даних?

Кількість станів можна визначити експериментально. Оскільки прихована вершина насправді діє як випадкова причина, то очікувано, що кількість станів буде відповідати кількості станів вершин, які вона розділяє. Насправді, кількість станів можна встановити як максимальне з числа станів предків і нащадків і потім видаляти стани, якщо вони матимуть в остаточному випадку досить низькі ймовірності.

Одними з перших дослідників проблеми знаходження умовних ймовірностей або навчання (оцінювання) параметрів прихованих вершин за умови відомої топології самої мережі, були Дж.-Л. Голмарда та А. Малет [40], які ще у 1991 описали алгоритм побудови мереж деревовидної структури. Пізніше

Д. Шпігельгалтером та Р. Коуеллом [41,42], К. Ольсеном, С. Лаурітценом, Ф. Дженсеном та ін. [42, 43] був досліджений загальний випадок для спрямованих ациклічних графів. У даних роботах описується алгоритм максимізації математичного очікування (ЕМ-алгоритм [44,45]) для ймовірнісних мереж. Деякі дослідники зазначали, що існують певні проблеми із застосуванням ЕМ-алгоритму для вирішення цієї проблеми, а тому запропонували, як можливу альтернативу, метод градієнтного спуску, який також дозволяє знайти локальний максимум на множині правдоподібності, що визначена параметрами множини. Активні дослідження щодо прямого порівняння обох методів тривають. Серед інших методів розв'язання проблеми навчання мереж Байєса з прихованими вершинами та відомою структурою варто виділити методи застосування вибірки Гілкса [46] та послідовного оновлення [47].

Процес навчання мережі Байєса складається із зовнішнього циклу, що є процесом пошуку структури, і внутрішнього циклу, що є процесом оптимізації параметрів. У разі повних даних внутрішній цикл виконується дуже швидко, оскільки при цьому досить лише витягнути інформацію про умовні частоти з набору даних. Якщо ж є приховані змінні, то для виконання внутрішнього циклу потрібно застосувати багато ітерацій алгоритму ЕМ або алгоритму на основі градієнта, а на кожній ітерації будуть обчислюватись розподіли апостеріорних ймовірностей в байєсівській мережі, що є NP-складною задачею.

### 2.3. Алгоритми навчання мереж Байєса з прихованими вершинами

*Методи на основі градієнтного підходу для навчання параметрів мереж Байєса*, що містять неспостережувані дані, були запропоновані у 1995 році Binder J., Koller D., Russell S. and Kanazawa K. [48-50]. Мета методу – знаходження оцінок параметрів мережі  $w_{ijk} = P(X_i = x_{ij} | Pa_i = pa_{ik})$ , максимізуючи логарифмічну функцію правдоподібності  $\ln P_w(D)$ . При цьому мають виконуватись наступні обмеження  $w_{ijk} \in [0,1]$ ,  $\sum_j w_{ijk} = 1$ . Алгоритм працює з вибіркою  $D$  (набори даних  $D_1 \dots D_M$ ), структурою мережі та початковими (випадково згенерованими) значеннями для таблиці умовних ймовірностей прихованих вершин.

Даний підхід ґрунтується на розгляді  $P_w(D)$  як функції від значень записів таблиць умовних ймовірностей  $w$ . Це зменшує проблему знаходження максимуму багатопараметричної нелінійної функції. Насправді, легше максимізувати функцію логарифма правдоподібності  $\ln P_w(D)$ . Оскільки ці дві функції монотонно зв'язані, то максимізація однієї еквівалентна максимізації іншої.

Найпростіший варіант даного підходу – це алгоритм градієнтного підйому (т. зв. «hill climbing» – пошук екстремуму). Для кожної точки  $w$  обчислюється  $\nabla w$  – градієнтний вектор частинних похідних за записами таблиці умовних ймовірностей. Потім алгоритм робить невеликий крок у напрямку градієнта до точки  $\nabla w$  – градієнтний вектор частинних похідних по записах таблиці умовних ймовірностей. Потім алгоритм робить невеликий крок в напрямку градієнта до точки  $w + \alpha \nabla w$ , де

$\alpha$  - розмір кроку. Цей простий алгоритм сходиться до локального оптимуму при достатньо малих  $\alpha$ .

Єдиною проблемою є те, що даний підхід має враховувати обмеження, що  $W$  складається зі значень умовних ймовірностей, тому  $w_{ijk} \in [0,1]$ . Більше того, у будь-якій таблиці умовних ймовірностей значення, що відповідають частковому умовному випадку ( залежно від стану предків) повинні в сумі давати 1, тобто  $\sum_j w_{ijk} = 1$ .

Один з методів врахування цих обмежень полягає у проектуванні  $\nabla W$  на множину обмежень. У загальному випадку даний тип проєкції виконується через проєкцію вектора градієнта на вектор, що є нормаллю до множини обмежень, і відніманням його від вихідного вектора градієнта. У цьому випадку маємо  $\sum_j w_{ijk} = 1$  для кожного  $i, k$ . Якщо для заданих  $i, k$  є декілька значень  $J$  для кожного  $j$ , вектор, ортонормальний до множини обмежень, матиме  $1/\sqrt{J}$  у кожній відповідній компоненті. Таким чином, проєкція градієнта на цей вектор матиме в кожній позиції  $ijk$  середнє значення компонентів градієнта, що відповідатиме  $w_{i1k}, \dots, w_{iJk}$ . Віднімаючи цей вектор від вихідного вектора градієнта, отримаємо перенормований вектор як результат.

У перенормованому векторі градієнту сума компонентів, що відповідають  $w_{ijk}$  при фіксованих  $i, k$  дорівнює нулю. Таким чином, якщо взяти малий крок за напрямком цього вектора, сума  $\sum_j w_{ijk}$  залишається незмінною.

Звідси, якщо почати роботу на множині обмежень, то обов'язково залишимося на цій множині, як і було задано. Алгоритм припиняє свою роботу, коли знайдений локальний максимум, тобто коли перенормований градієнт дорівнює нулю.

Ефективність використання градієнтних методів залежить від вибору ефективного методу обчислення градієнтів. Це є одним із ключових моментів успіху градієнтного спуску в нейронних мережах. Там зворотне поширення похибки (back-propagation) використовується для обчислення градієнта функції похибки по відношенню до параметрів мережі (параметрів зв'язків). Існування локального навчального алгоритму, що використовує результати виводу, спрощує проектування і реалізацію систем на основі нейронних мереж.

Схожа ситуація спостерігається і в байєсівських мережах. Фактично, для ймовірнісних мереж алгоритм виводу вже об'єднує всі необхідні обчислення, тому немає необхідності у будь-якому додатковому зворотному поширенні похибки. Градієнт може бути обчислений локально для кожної вершини з використанням інформації, що доступна з нормальних обчислень мережі Байєса.

Обчислимо вклад до градієнта кожного набору даних окремо і потім просумуємо результат:

$$\frac{\partial \ln P_W(D)}{\partial w_{ijk}} = \frac{\partial \ln \prod_{l=1}^m P_W(D_l)}{\partial w_{ijk}} = \sum_{l=1}^m \frac{\partial \ln P_W(D_l)}{\partial w_{ijk}} = \sum_{l=1}^m \frac{\partial P_W(D_l) / \partial w_{ijk}}{\partial P_W(D_l)}, \quad (2.4)$$

Знайдемо вираз для обчислення  $\sum_{l=1}^m \frac{\partial P_w(D_l) / \partial w_{ijk}}{\partial P_w(D_l)}$ .

$$\frac{\partial P_w(D_l) / \partial w_{ijk}}{\partial P_w(D_l)} = \frac{\frac{\partial}{\partial w_{ijk}} \left( \sum_{j',k'} P_w(D_l | x_{ij'}, u_{ik'}) P_w(x_{ij'}, u_{ik'}) \right)}{P_w(D_l)} =$$

$$= \frac{\frac{\partial}{\partial w_{ijk}} \left( \sum_{j',k'} P_w(D_l | x_{ij'}, u_{ik'}) P_w(x_{ij'} | u_{ik'}) P_w(u_{ik'}) \right)}{P_w(D_l)}, \quad (2.5)$$

Як видно з формули (2.5)  $w_{ijk}$  з'являється лише в лінійній формі. Фактично,  $w_{ijk}$  з'являється лише в одному члені суми, для якого  $j' = j, k' = k$ . Для цього випадку  $P_w(x_{ij'} | u_{ik'}) = w_{ijk}$ . Звідси,

$$\frac{\partial P_w(D_l) / \partial w_{ijk}}{\partial P_w(D_l)} = \frac{P_w(D_l | x_{ij}, pa_{ik}) P_w(pa_{ik})}{P_w(D_l)} = \frac{P_w(x_{ij}, pa_{ik} | D_l) P_w(D_l) P_w(pa_{ik})}{P_w(x_{ij}, u_{ik}) P_w(D_l)} =$$

$$= \frac{P_w(x_{ij}, pa_{ik} | D_l)}{P_w(x_{ij}, | pa_{ik})} = \frac{P_w(x_{ij}, pa_{ik} | D_l)}{w_{ijk}} \quad (2.6)$$

Рівняння (2.5) дозволяє обчислювати градієнт разом з обчисленням апостеріорних ймовірностей з використанням звичайних операцій з мережами Байєса. По суті, будь-який стандартний алгоритм виводу для мереж Байєса для набору даних  $D_l$  обчислить значення  $P_w(x_{ij}, pa_{ik} | D_l)$  як побічний продукт.

Вектор градієнту тепер може бути отриманий таким чином. Використовуючи алгоритм ймовірнісного висновку для кожного набору даних, окремо обчислюються  $P_w(x_{ij}, pa_{ik} | D_l)$  для кожного  $ijk$ . Потім ці значення додаються для всіх сукупностей даних і діляться на  $w_{ijk}$ .

Алгоритм має вигляд:

**repeat until**  $\Delta w \approx 0$

$\Delta w \leftarrow 0$

**for each**  $D_i \in D$

ймовірнісний висновок в мережі по  $D_l$

for each змінної  $i$ , стану  $j$ , умовного стану  $k$

$$\Delta w_{ij} \leftarrow \Delta w_{ijk} + \frac{\partial \ln P_w(D)}{\partial w_{ijk}} = \frac{P_w(x_{ij}, pa_{ik} | D_l)}{w_{ijk}}$$

$\Delta w_{ijk} \leftarrow$  проекція на простір обмежень

$w \leftarrow w + \alpha \Delta w$



Таким чином, отримано алгоритм навчання мереж Байєса, що містять приховані змінні. Він має назву **APN** (adaptive probabilistic network) алгоритм. Таку мережу називають адаптивною ймовірнісною мережею (adaptive probabilistic network або скорочено APN).

**Алгоритм expectation maximization (EM) застосовується**, коли деякі вершини мережі Байєса приховані, а структура відома для знаходження локально оптимальних оцінок максимальної правдоподібності для параметрів. EM-алгоритм або алгоритм максимізації математичного очікування, запропонований ще у 1977 році А. П. Демпстером, Н. М. Лаердом та Д. Б. Рубіном [44] і адаптований для навчання мереж Байєса з прихованими вершинами та відомою структурою С. Л. Лауритценом [51].

Алгоритм складається з двох базових кроків  $E$  (expectation) та  $M$  (maximization). Головна ідея алгоритму полягає в тому, що якби були відомі значення всіх вершин, то навчання (на кроці  $M$ ) було б простим, оскільки була б наявна вся необхідна інформація. Тому спочатку на кроці  $E$  робиться обчислення значення очікуваної правдоподібності (expectation of the likelihood) включаючи латентні змінні, так ніби є можливість їх спостерігати. Фактично робиться “доповнення” даних для прихованих змінних на основі поточних оцінок параметрів мережі  $\theta^{(i)}$ , тобто знаходиться розподіл апостеріорних ймовірностей по прихованих змінних за наявності отриманих даних. На кроці  $M$  обчислюються значення оцінок максимальної правдоподібності для параметрів, використовуючи максимізацію значень очікуваної правдоподібності, отриманих на кроці  $E$ . Далі алгоритм рекурсивно виконує крок  $E$  з використанням параметрів  $\theta^{(i+1)}$ , отриманих на кроці  $M$ , і так далі.

Нехай  $X = \{X_1, \dots, X_n\}$  - вершини мережі Байєса;  
 $\Theta = \{\theta_{ijk}\}, 1 \leq i \leq n, 1 \leq j \leq q_i, 1 \leq k \leq r_i$  - множина таблиць умовних ймовірностей  $P(X_i = x_k | Pa(X_i) = pa_j)$  (де  $q_i$  та  $r_i$  кількість станів вершини  $X_i$  та її предків  $Pa(X_i)$  відповідно);  $D = \{X_i^l\}, 1 \leq i \leq n, 1 \leq l \leq N$  - множина даних, а  $D_m$  - дані прихованих змінних, які відсутні в навчальній вибірці.

Нехай  $\log P(D | \Theta)$  - логарифмічна функція правдоподібності параметрів при заданому наборі даних. Встановивши певні початкові значення  $\Theta^*$ , можна обчислити ймовірнісний розподіл для прихованих змінних  $P(D_m | \Theta^*)$  і потім обчислити  $Q(\Theta : \Theta^*)$  - математичне очікування попередньої логарифмічної функції правдоподібності:

$$Q(\Theta : \Theta^*) = E_{\Theta^*} \log P(D | \Theta), \quad (2.7)$$

Рівняння (2.7) може бути записане в наступному вигляді:

$$Q(\Theta : \Theta^*) = \sum_i \sum_{X_i=x_k} \sum_{P_i=pa_j} N_{ijk}^* \log P_{\theta_{ijk}}(D), \quad (2.8)$$

де  $N_{ijk}^* = E_{\Theta^*}[N_{ijk}] = N \cdot P(X_i = x_k, P_i = pa_j | \Theta^*)$  отримується за допомогою

ймовірнісного висновку в мережі Байєса.

**Схема EM-алгоритму:**

*Крок 1:  $i=0$*

*вибір*

*Крок 2: repeat*

*Крок 3:  $i=i+1$*

*Крок 4:  $\Theta^i = \arg_{\theta} \max Q(\Theta : \Theta^{i-1})$*

*Крок 5: until  $Q(\Theta : \Theta^{i-1}) \leq Q(\Theta^{i-1} : \Theta^{i-1})$*

Таким чином, на кроці  $E$  оцінюється значення  $N^*$  на основі параметрів  $\Theta^i$ , а на кроці  $M$  вибирається найкраще значення параметрів  $\Theta^{i+1}$ , максимізуючи  $Q$ .

У роботі [44] було доведено, що алгоритм сходиться, а також факт того, що не обов'язково буде знайдено глобальний оптимум функції  $Q(\Theta : \Theta^*)$ . Наявність локальних оптимумів є серйозним недоліком даного алгоритму. Одне з рішень полягає в накладенні розподілів апіорної вірогідності на параметри моделі і застосуванні версії MAP (Maximum a posteriori) алгоритму EM. Ще одне рішення полягає у перезапуску обчислень з новими, випадково вибраними параметрами. Іноді також має сенс вибирати для ініціалізації параметрів більш обґрунтовані значення.

На основі розглянутого принципу EM-алгоритму можливо визначити його певні варіанти і удосконалення. Наприклад, дуже часто E-крок (обчислення розподілів апостеріорних ймовірностей по прихованих змінних) досить трудомісткий у випадку побудови великих байєсових мереж. У цьому випадку можна застосувати наближений E-крок, і алгоритм навчання буде залишатись ефективним. При використанні алгоритму формування вибірки за методом Монте-Карло для марковських ланцюгів (МКМЛ) процес навчання стає повністю інтуїтивно зрозумілим: кожен стан (конфігурація прихованих і спостережуваних змінних), який згенерував алгоритм за методом Монте-Карло для марковських ланцюгів, розглядається точно так, ніби він був повним спостереженням. Тому параметри можуть оновлюватися безпосередньо після кожного переходу із стану в стан в алгоритмі за методом Монте-Карло для марковських ланцюгів. При оцінюванні параметрів дуже великих мереж шляхом навчання також показали свою високу ефективність інші методи формування наближеного ймовірнісного висновку – варіаційні та циклічні.

Якщо ж доводиться враховувати приховані змінні, то завдання ускладнюється. У простому випадку приховані змінні можуть бути включені в загальний список разом із спостережуваними змінними; хоча їх значення не спостерігаються, алгоритм навчання отримує відомості про те, що вони існують, і повинен знайти для них місце в структурі мережі. Наприклад, за допомогою алгоритму може бути зроблена спроба визначити в процесі навчання структуру, показану на рис. 2.2 на основі тієї інформації, що в модель має бути включена змінна HeartDisease (тризначна змінна).

Якщо ж алгоритму навчання не надана ця інформація, то в процесі навчання виникають два варіанти: або виходити з того, що дані дійсно є повними (що змушує алгоритм визначити в процесі навчання модель з набагато більшою кількістю

параметрів, показану на рис. 2.2, б), або винайти нові приховані змінні для спрощення моделі. Останній підхід можна реалізувати шляхом включення нових варіантів операцій модифікації в процедуру пошуку структури - передбачити в алгоритмі можливість не лише модифікувати зв'язки, але і додавати або видаляти приховані змінні, або міняти їх арність.

Безумовно, в такому алгоритмі не можна передбачити можливість назвати знову винайдену змінну, допустимий HeartDisease, а також присвоювати її значенням осмислені імена. Але знову винайдені приховані змінні зазвичай зв'язуються з раніше існуючими змінними, тому людина - фахівець у цій проблемній області часто може перевіряти локальні розподіли умовної вірогідності, що стосуються нової змінної, і встановлювати її сенс.

Розглянемо випадок, зображений на рис. 2.3, коли є два пакети з цукерками, вміст яких згодом змішали. Для опису цукерок застосовано три характеристики: окрім різновиду Flavor і обгортки Wrapper, застосовано ще характеристику "з отворами" (Hole), оскільки частина цукерок – льодяники з отворами (Hole) всередині, а деякі – льодяники без отворів.

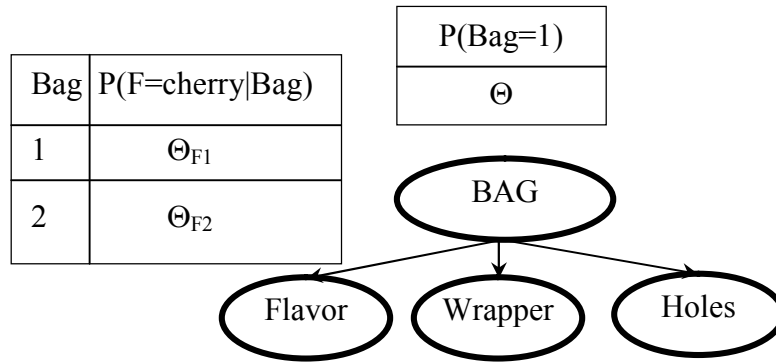


Рис. 2.3. Наївна мережа Байеса для ситуації розподілу цукерок по пакетах

Розподіл цукерок у кожному пакеті описано за допомогою наївної байєсівської моделі: у кожному конкретному пакеті характеристики незалежні між собою, але розподіл умовних ймовірностей кожної характеристики залежить від пакету. При цьому застосовано такі параметри:  $\theta$  - апіорна ймовірність того, що цукерка узята з пакета Bag1;  $\theta_{F1}$  і  $\theta_{F2}$  - ймовірності того, що цукерка є вишневим льодяником, за умови, що вона взята з пакету Bag<sub>1</sub> і Bag<sub>2</sub> відповідно;  $\theta_{W1}$  і  $\theta_{W2}$  задають ймовірності того, що обгортка має червоний колір; а  $\theta_{H1}$  і  $\theta_{H2}$  - ймовірності того, що льодяник має отвір.

На рис. 2.3 прихована змінна відповідає просто пакету, оскільки після змішування цукерок немає можливості визначити, з якого пакету була взята кожна цукерка. Завдання полягає у відновленні опису цих двох пакетів, маючи лише характеристики цукерок, узятих з цієї суміші. Проведемо одну ітерацію алгоритму ЕМ для розв'язання цієї задачі. Спочатку випадково згенеруємо 1000 вибірок з моделі, істинними параметрами якої є такі:

$$\theta = 0,5, \theta_{F1} = \theta_{W1} = \theta_{H1} = 0,8, \theta_{F2} = \theta_{W2} = \theta_{H2} = 0,2.$$

Це означає рівноймовірне отримання цукерок з одного або іншого пакету: у

першому пакеті в основному знаходяться вишневі льодяники в червоних обгортках і з отворами, а в другому – в основному лимонні льодяники в зелених обгортках і без отворів. Кількість цукерок восьми можливих різновидів визначена у таблиці 2.2.

Таблиця 2.2

Кількості можливих різновидів цукерок

	W = red		W = green	
	H = 1	H = 0	H = 1	H = 0
F = cherry	273	93	104	90
F = lime	79	100	94	167

Функціонування EM-алгоритму розпочинається з початкової ініціалізації параметрів. Наприклад, виберемо такі значення параметрів:

$$\theta^{(0)} = 0,6, \theta_{F1}^{(0)} = \theta_{W1}^{(0)} = \theta_{W1}^{(0)} = 0,6, \theta_{F2}^{(0)} = \theta_{W2}^{(0)} = \theta_{H2}^{(0)} = 0,4$$

Обчислимо оцінку параметра  $\theta$ . У випадку повних спостережень можна було б отримати оцінку для цього параметра безпосередньо із спостережуваних значень кількості цукерок, що відносяться до пакетів 1 і 2. Але оскільки номер пакету – це прихована змінна, розрахуємо замість цього очікувані значення кількості. Очікувана кількість  $N(\text{Bag}=1)$  є сумою ймовірностей того, що цукерка узята з пакету 1, по усіх цукерках (2.9):

$$\theta^{(1)} = \hat{N}(\text{Bag} = 1) / N = \sum_{j=1}^N P(\text{Bag} = 1 | \text{flavor}_j, \text{wrapper}_j, \text{holes}_j) / N. \quad (2.9)$$

Цю ймовірність можна обчислити за допомогою будь-якого алгоритму імовірного виводу для байєсових мереж. А в даному випадку, наприклад, можна використати правило Байєса і застосувати вираз для умовної незалежності:

$$\theta^{(1)} = \frac{1}{N} \sum_{j=1}^N \frac{P(\text{flavor}_j | \text{Bag}=1) P(\text{wrapper}_j | \text{Bag}=1) P(\text{holes}_j | \text{Bag}=1) P(\text{Bag}=1)}{\sum_{i=1}^2 P(\text{flavor}_j | \text{Bag}=i) P(\text{wrapper}_j | \text{Bag}=i) P(\text{holes}_j | \text{Bag}=i) P(\text{Bag}=i)}. \quad (2.10)$$

Варто звернути увагу на те, що нормуюча константа також залежить від параметрів. Зробивши розрахунки для всіх видів цукерок, кількість яких вказана у таблиці 2.3, отримаємо, що  $\theta^{(1)} = 0.6124$ .

Розглянемо процес оцінювання інших параметрів, наприклад,  $\theta_{F1}$ . У випадку повних спостережень, це значення можна було б оцінити безпосередньо на основі спостережуваних значень кількості вишневих і лимонних льодяників з пакету 1. Очікувана кількість вишневих льодяників з пакету 1 обчислюється за допомогою наступного виразу:

$$\sum_{j: \text{Flavor}_j = \text{cherry}} P(\text{Bag} = 1 | \text{Flavor}_j = \text{cherry}, \text{wrapper}_j, \text{holes}_j).$$

Цю ймовірність також можна обчислити за допомогою будь-якого алгоритму для байєсової мережі.

Продовжуючи цей процес, знайдемо оцінки для всіх параметрів:

$$\theta^{(1)} = 0,6124, \theta_{F1}^{(1)} = 0,6684, \theta_{W1}^{(1)} = 0,6483, \theta_{H1}^{(1)} = 0,6558,$$

$$\theta_{F2}^{(1)} = 0,3887, \theta_{W2}^{(1)} = 0,3817, \theta_{H2}^{(1)} = 0,3827.$$

Логарифмічна правдоподібність даних зростає від первинного значення, приблизно рівного, -2044, приблизно до -2021 після першої ітерації, як показано на рис. 2.4.

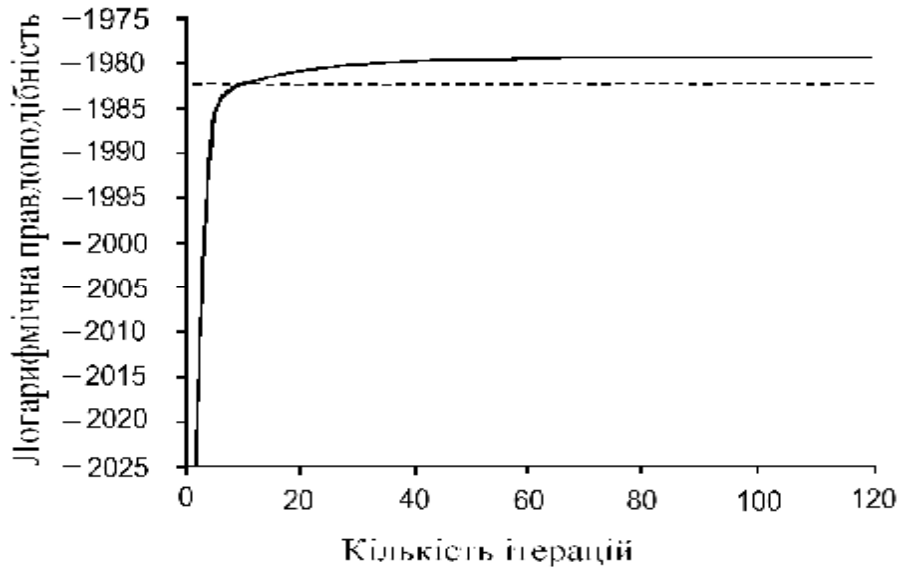


Рис. 2.4. Графік зміни значення логарифмічної правдоподібності залежно від кількості ітерацій EM-алгоритму

До десятої ітерації модель, отримана у процесі навчання, краще відповідає даним, ніж первинна модель ( $L = -1982,214$ ). Але подальший прогрес значно сповільнюється. Така ситуація в застосуваннях алгоритму EM зустрічається дуже часто, тому в багатьох практично вживаних системах алгоритм EM на останньому етапі навчання використовується у поєднанні з таким алгоритмом на основі градієнта, як алгоритм Ньютона-Рафсона.

Загальний висновок, який можна зробити на підставі цього прикладу, полягає в тому, що оновлення параметрів при навчанні байєсової мережі з прихованими змінними є безпосередньо доступним за допомогою результатів формування імовірнісного висновку для кожного прикладу. Більше того, для кожного параметра необхідно мати тільки локальні апостеріорні ймовірності. Для загального випадку, у якому в процесі навчання оцінюються параметри умовних ймовірностей для кожної змінної  $X_i$ , якщо дані її батьківські змінні, інакше кажучи,  $\theta_{ijk} = P(X_i = x_{ij} | Pa_i = pa_{ik})$ , оновлення задається за допомогою нормалізованих очікуваних кількостей таким чином:

$$\theta_{ijk} \leftarrow \hat{N}(X_i = x_{ij}, Pa_i = pa_{ik}) / \hat{N}(Pa_i = pa_{ik}).$$

Ці очікувані величини можна отримати шляхом обчислення сум для усіх прикладів і обчислення ймовірностей  $P(X_i = x_{ij}, Pa_i = pa_{ik})$  для кожного з них з використанням будь-якого алгоритму імовірнісного висновку в мережах Байєса.

## 2.4. Особливості побудови мережі Байєса з прихованими вершинами та невідомою структурою

Одним із найскладніших випадів є побудова мереж Байєса з прихованими вершинами та невідомою структурою. Структурне навчання полягає у знаходженні правильних зв'язків між вершинами. Цікавішою проблемою є створення прихованих вершин. Приховані вершини можуть зробити модель компактнішою. Стандартний підхід полягає в додаванні щоразу по одній прихованій вершині до певної частини мережі, виконуючи структурне навчання на кожному кроці до тих пір, поки оцінка моделі не перестане зменшуватись. Ще одна проблема – це вибір можливої кількості прихованих вершин та типу розподілу їх умовних ймовірностей. Разом з цим стоїть задача вибору місця створення прихованої вершини. Немає сенсу робити її нащадком, оскільки приховані нащадки завжди можуть бути скорочені. Тому необхідно знайти існуючу вершину, яка потребує нового предка у випадку, коли поточна множина можливих предків недостатньо адекватна.

Одним з підходів може бути дослідження умовної ентропії  $H(X | Pa(X))$ . Якщо вона дуже велика, то це значить, що поточна множина не може "пояснити" залишкову ентропію; якщо  $Pa(X)$  найкраща множина предків (за значенням ВІС (Bayesian Information Criteria) або хі-квадрат), яку можливо знайти в поточній моделі, то це передбачає необхідність створення нової вершини і включення її до множини  $Pa(X)$ .

Для розв'язання завдання **структурного навчання мережі Байєса** з прихованими вершинами використовують **структурний EM-алгоритм** або алгоритм стиснення границь.

Одним з удосконалень вищеописаного принципу максимізації математичного очікування є структурний EM-алгоритм [52] (SEM – structural EM algorithm), який діє багато у чому так само, як і звичайний (параметричний) алгоритм EM, за винятком того, що цей алгоритм може оновлювати не лише параметри, але і структуру моделі.

**Схема алгоритму SEM має вигляд:**

**Крок 1:**  $i=0$

**вибір початкової мережі**  $(G^0, \Theta^0)$

**Крок 2:** *repeat*

**Крок 3:**  $i=i+1$

**Крок 4:**  $(G^i, \Theta^i = \arg_{G, \theta} \max Q(G, \Theta : G^{i-1}, \Theta^{i-1})$

**Крок 5:** *until*  $Q(G^i, \Theta^i : G^{i-1}, \Theta^{i-1}) \leq Q(G^{i-1}, \Theta^{i-1} : G^{i-1}, \Theta^{i-1})$

Цей алгоритм навчає мережі на основі штрафних ймовірнісних значень, які включають значення, отримані за допомогою байєсівського інформаційного критерію (ВІС), принципу мінімальної довжини (MDL), а також інших критеріїв. Як і у звичайному алгоритмі EM, тут використовуються поточні параметри для обчислення очікуваних величин на E-кроці, а потім ці величини застосовують на M-кроці для вибору нових параметрів. В алгоритмі структурного EM використовується структура для обчислення очікуваних величин, після чого ці величини застосовують на M-

кроці для оцінювання правдоподібності нових потенційних структур (у цьому полягає відмінність цього методу від методу зовнішнього циклу/внутрішнього циклу, в якому обчислюються нові очікувані величини для кожної потенційної структури). Крок 4 даного алгоритму передбачає пошук найкращої структури та найкращої множини параметрів для даної структури. На практиці ці кроки чітко відрізняються:

$$G^i = \arg_G \max Q(G, \Theta^i : G^{i-1}, \Theta^{i-1}), \quad (2.13)$$

$$\Theta^i = \arg_{\theta} \max Q(G^i, \Theta : G^{i-1}, \Theta^{i-1}). \quad (2.14)$$

Перший пошук (2.13) у просторі можливих графів повертає до вихідної проблеми пошуку оптимальної структури мережі Байєса. Щодо пошуку у просторі параметрів (2.14), Н. Фрідменом [45] запропоновано повторювати цю процедуру декілька разів з використанням інтелектуальної ініціалізації. Даний крок означає запуск параметричного ЕМ-алгоритму для кожної структури  $G^i$ , починаючи із структури  $G^0$ .

Таким чином, структурний алгоритм ЕМ дає можливість вносити до мережі декілька структурних змін без повторних обчислень очікуваних величин і має здатність визначати у процесі навчання нетривіальні структури байєсівських мереж. Проте необхідно виконати ще багато дослідницької роботи, перш ніж можна буде стверджувати, що завдання визначення структури в процесі навчання вирішене остаточно.

Структурний ЕМ-алгоритм дає можливість оцінювати параметри і структури мережі Байєса з визначеною кількістю прихованих змінних. Однак проблема визначення кількості прихованих вершин, які необхідно додавати до структури мережі, на сьогодні остаточно не вирішена. Тому існує гостра необхідність у створенні ефективних додаткових механізмів для розв'язання цієї задачі.

Ще одним методом структурного навчання мереж Байєса з прихованими вершинами є **алгоритм стиснення границь** (bound and collapse), який запропоновано і описано в роботах М. Рамони та П. Себастьяні [53, 54].

Метод стиснення границь моделює відсутність даних, припускаючи що ймовірність даних, які відсутні, приймає значення в інтервалі від 0 до 1. Тобто виконується обчислення цього інтервалу на відсутніх даних за наявною інформацією. Після цього виконується стиснення границь інтервалу в точку за допомогою використання опуклої комбінації з точок екстремумів, використовуючи інформацію про неповні дані.

## 2.5. Методика знаходження параметрів байєсівських мереж з прихованими вершинами

На основі розглянутого вище алгоритму максимізації математичного очікування розроблено методику обчислення параметрів мережі Байєса за умови неповної вхідної інформації та відомої топології мережі. Отримана методика повністю описує весь процес оцінювання невідомих параметрів прихованих вершин і складається з таких кроків:

- 1) побудова мережі Байєса за навчальними даними або “вручну”;
- 2) генерування вибірки по заданій структурі мережі (застосовується випадку відсутності навчальних даних);
- 3) додавання прихованих вершин до структури мережі;
- 4) початкова ініціалізація невідомих параметрів мережі;
- 5) обчислення оцінок параметрів мережі на основі згенерованих даних з використанням алгоритму ЕМ.

Схематичне зображення запропонованої методики показано на рис. 2.6.

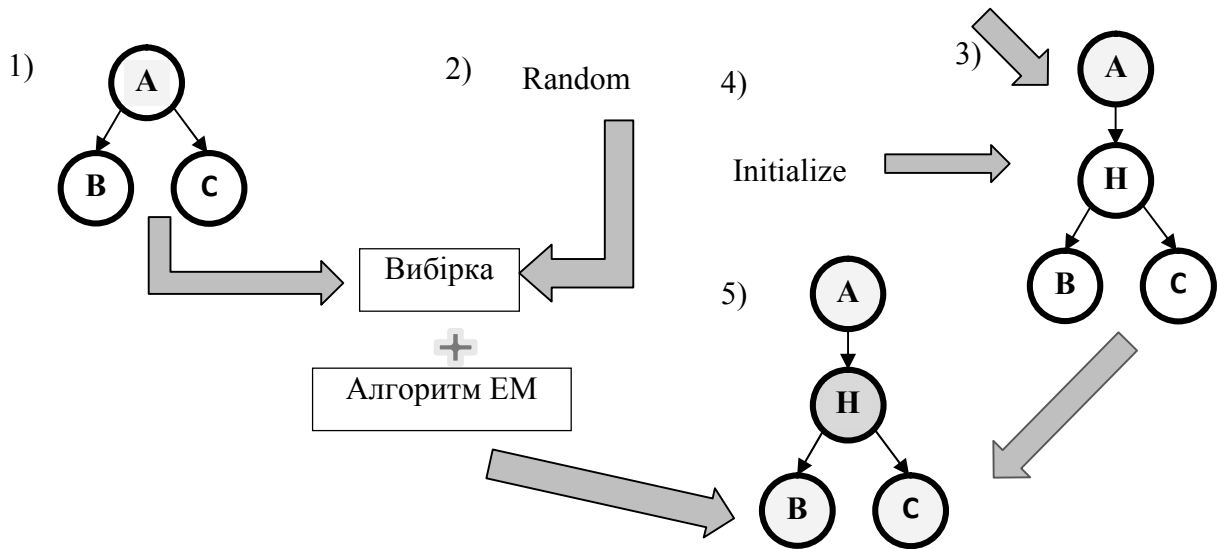


Рис. 2.6. Схема методики знаходження параметрів мережі Байєса з прихованими вершинами

На першому етапі необхідно побудувати мережу Байєса, тобто визначити топологію і знайти значення параметрів всієї мережі. На цьому етапі можливі *два варіанти*:

- 1) структура мережі нам відома і тоді залишається перенести її у програмне середовище і заповнити таблиці умовних ймовірностей;
- 2) є лише навчальні дані.

У другому випадку побудова структури здійснюється за два кроки: на першому кроці будується топологія мережі з використанням, наприклад евристичного алгоритму, а на другому обчислюють оцінки параметрів мережі, які максимально правдоподібні до навчальних даних.

На другому етапі у випадку, коли навчальні дані не задані або їх недостатньо, відбувається генерування псевдовипадкової вибірки за побудованою на першому етапі структурою мережі. Генерування відбувається таким чином. Спочатку формується ймовірнісний висновок у мережі без інстанційованих вершин  $P(S_{ij})$ ,  $i = 1, \dots, N$ ;  $j = 1, \dots, S$ . Далі обирається вершина  $N_{i^*}$  та інстанціюється один із її станів  $S_{i^*j^*}$  із ймовірністю цього стану  $P(S_{i^*j^*})$ . Потім перераховуються ймовірності станів вершин після інстанціювання  $P(S_{ij} | S_{i^*} = S_{i^*j^*})$ , а далі обирається наступна



вершина. Ця операція повторюється доти, доки не залишиться неінстанційованих вершин. Інстанційовані стани утворюють запис у вибірці  $(S_{i_1} \dots S_{i_N})$ ,  $i_k \in (1, \dots, S)$ .

Алгоритм повторюється до тих пір, доки не буде створено необхідної кількості записів у вибірці.

На третьому етапі відбувається додавання прихованої вершини до мережі. Якщо ця вершина вставляється між декількома існуючими, то видаляються попередні дуги між ними і створюються нові, які пов'язані з прихованим вузлом. Якщо необхідно додати приховану батьківську вершину, то просто створюється нова вершина і відповідні дуги.

На наступному етапі параметри прихованих вершин ініціалізуються початковими значеннями. Це можуть бути як випадково згенеровані значення, так і значення, які надаються експертом.

На завершальному етапі запускається ітераційний процес алгоритму EM, який використовує попередньо згенеровану вибірку даних і результатом якого є оцінки невідомих параметрів прихованих вершин мереж Байєса.

## 2.6. Приклади практичного застосування мереж Байєса з прихованими вершинами

В якості прикладів практичного застосування мереж з прихованими вершинами, розглянемо класичні байєсівські мережі: Asia, Alarm, Car Starts.

Ці мережі входять до стандартного набору прикладів майже всіх відомих програмних продуктів для побудови і застосування мереж Байєса. Вони є еталонними мережами Байєса, які застосовуються дослідниками для тестування нових та існуючих алгоритмів побудови мереж та ймовірнісного висновку.

Процес тестування побудованих мереж за використання еталонних виконується наступним чином. Для тестування використовується або мережа повністю, або певна її частина (у випадку великої мережі для підвищення швидкості навчання). Видаляється, а потім знову додається вершина з деякими початковими значеннями для таблиці умовних ймовірностей тієї вершини, яка має бути прихованою і виконується алгоритм навчання параметрів прихованих вершин.

Результати роботи програми аналізуються таким чином: порівнюються істинні значення ймовірностей станів прихованої вершини, а також таблиць умовних ймовірностей зі значеннями, отриманими в результаті проведення експерименту. Також досліджується залежність логарифмічної функції правдоподібності мережі від кількості ітерацій, виконаних за алгоритмом максимізації математичного очікування.

**Мережа Alarm** – A Logical Alarm Reduction Mechanism. Мережа Alarm була описана і вперше використана при дослідженні мереж Байєса у 1989 році в роботі [55]. Ця мережа містить 37 вершин та 46 дуг. Вона описує систему повідомлення тривоги для контролю стану пацієнтів лікарні; при цьому обчислюються ймовірності для диференційного діагнозу на основі доступних даних. Медичні знання закодовані у графічній структурі, яка пов'язує 8 діагнозів, 16 відомостей та 13 проміжних змінних. Топологію мережі зображено на рис. 2.7.

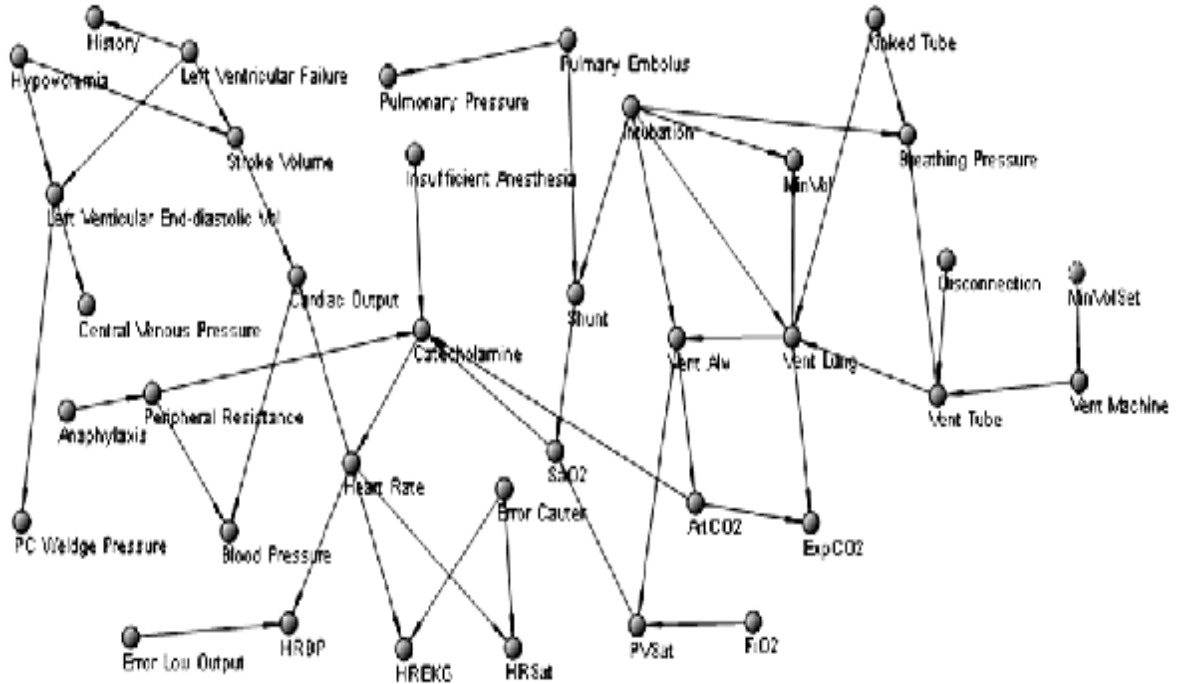


Рис. 2.7. Топологія мережі "Alarm"

Оскільки мережа Alarm є досить великою за кількістю вершин, то для використання методики знаходження невідомих параметрів прихованих вершин необхідно мати значні обчислювальні ресурси.

Для зменшення часу функціонування тесту розглянуто лише фрагмент мережі, над яким був проведено відповідний експеримент.

Цей фрагмент зображено на рис. 2.8, прихованою є вершина Left Ventricular End-diagnostic Vol.

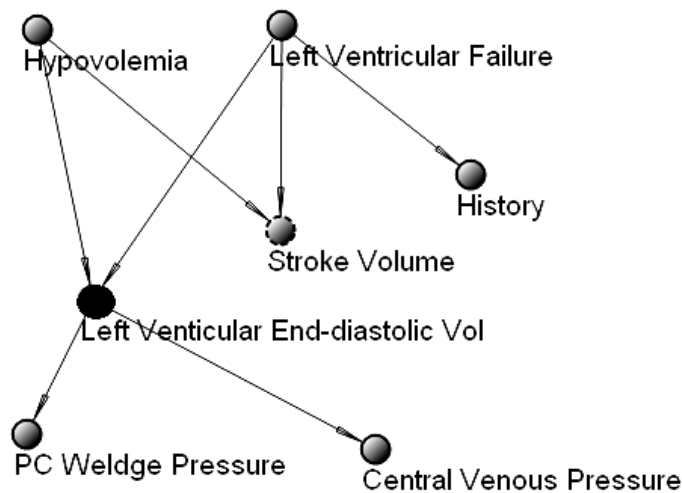


Рис. 2.8. Фрагмент мережі Alarm з введенням прихованої вершини

У таблиці 2.4 наведено істинні значення ймовірностей станів прихованої вершини та оцінки цих значень, отримані за допомогою програми BNetMaster. Як бачимо, ці значення є дуже близькими між собою.

Таблиця 2.4

Значення ймовірностей станів прихованої вершини

Стан	Початкові значення	Експериментальні оцінки
Low	0,230	0,251
Normal	0,690	0,694
High	0,076	0,055

У таблицях 2.5 та 2.6 наведено умовні ймовірності прихованої вершини: відповідно, початкова та експериментально обчислена. Ці значення є також дуже близькими між собою.

Таблиця 2.5

Істинна таблиця умовних ймовірностей прихованої вершини

Hypovolemia	LVFailure	Low	Normal	High
TRUE	TRUE	0,95	0,04	0,01
TRUE	FALSE	0,98	0,01	0,01
FALSE	TRUE	0,01	0,09	0,9
FALSE	FALSE	0,05	0,9	0,05

Таблиця 2.6

Експериментально обчислена таблиця умовних ймовірностей прихованої вершини

Hypovolemia	LVFailure	Low	Normal	High
TRUE	TRUE	0,99	0,01	0
TRUE	FALSE	0,98	0,01	0,01
FALSE	TRUE	0,01	0,23	0,76
FALSE	FALSE	0,06	0,9	0,04

На рис. 2.9 зображено залежність логарифмічної функції правдоподібності від кількості ітерацій алгоритму EM.

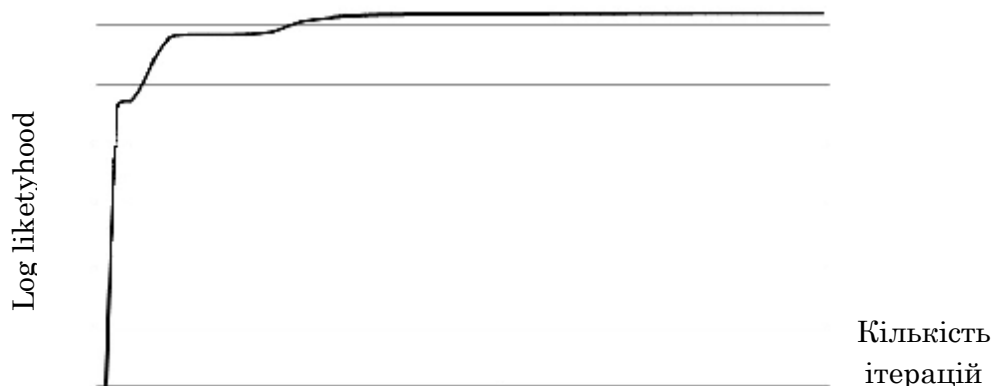


Рис. 2.9. Логарифмічна функція правдоподібності для мережі "Alarm"

З рис. 2.9 видно, що з кожною ітерацією значення функції правдоподібності зростає, що свідчить про покращення параметрів мережі. У цілому за весь процес навчання логарифм функції правдоподібності збільшився з -5,47 до -1,90.

**Мережа Asia** [56] - це спрощена версія мережі Байєса, що може використовуватись для діагностування пацієнта. Так визначається, чи хворіє пацієнт на туберкульоз ("Tuberculosis"), рак легень ("LungCancer") чи бронхіт ("Bronchitis") залежно від його рентгенівського знімку ("X-Ray"), наявності задишки ("Dyspnea"), перебування в Азії ("Visit to Asia") та паління ("Smoker"). Топологія мережі представлена на рис. 2.10.

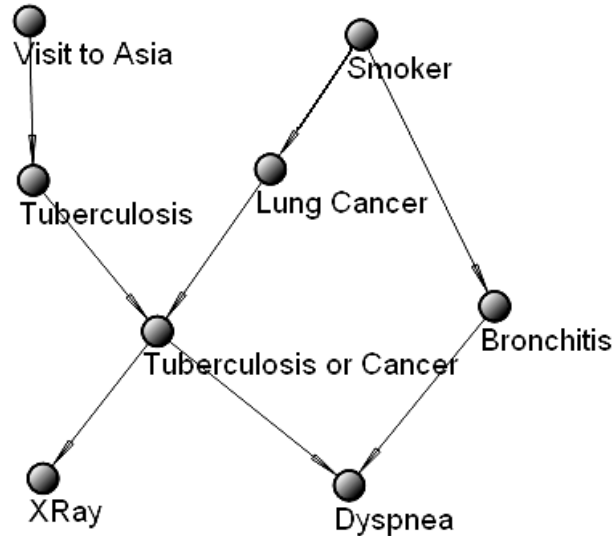


Рис. 2.10. Топологія мережі Asia

Мережа "Asia" містить 8 вершин та 8 дуг. Кожна вершина мережі відповідає певному стану пацієнта, наприклад, "Visit to Asia" показує чи відвідував пацієнт Азію останнім часом.

Ще одна назва мережі – "Chest Clinic". В якості прихованої вибиралась вершина "Tuberculosis or Cancer".

У таблиці 2.5 наведено реальні значення ймовірностей станів прихованої вершини та оцінки цих значень, отримані в результаті роботи програми VNetMaster. Похибка між значеннями є незначною.

Таблиця 2.5

Значення ймовірностей станів прихованої вершини

Стан	Початкові значення	Експериментальні оцінки
True	0,065	0,054
False	0,935	0,946

У таблицях 2.6 та 2.7 наведено умовні ймовірності прихованої вершини: відповідно, початкова та експериментально обчислена.

Отримані значення повторюють фактичні.

Таблиця 2.6

Істинна таблиця умовних ймовірностей прихованої вершини

Tuberculosis	LungCancer	True	False
Present	present	100	0
Present	absent	0	100
Absent	present	100	0
Absent	absent	0	100

Таблиця 2.7

Експериментально обчислена таблиця умовних ймовірностей прихованої вершини

Tuberculosis	Lung Cancer	True	False
Present	present	0,99	0,01
Present	absent	0,01	0,99
Absent	present	100	0
Absent	absent	0,01	0,99

На рис. 2.11 зображено залежність логарифмічної функції правдоподібності від кількості ітерацій алгоритму EM.

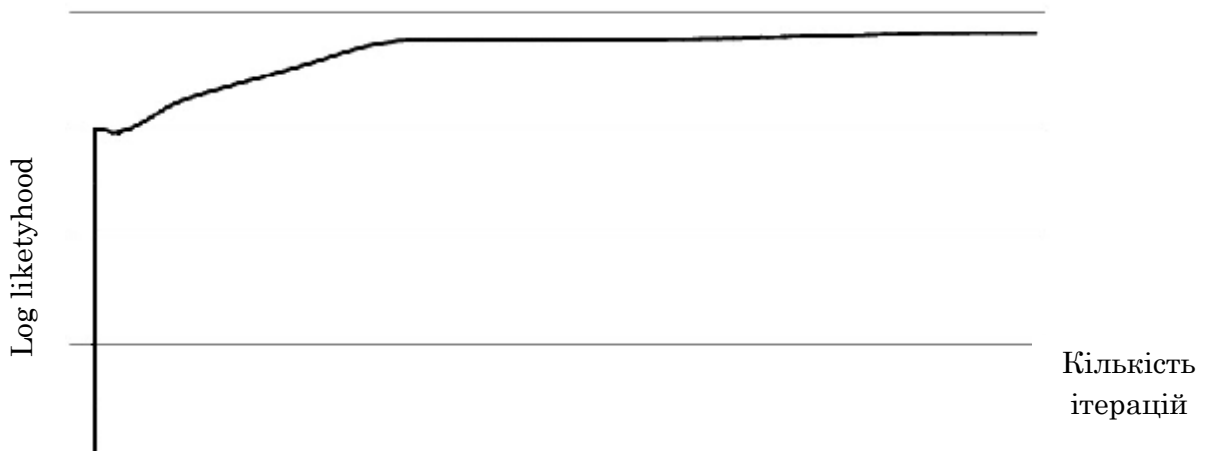


Рис. 2.11 Графік логарифмічної функції правдоподібності для мережі "Asia"

З рис. 2.11 видно, що з кожною ітерацією значення функції правдоподібності зростає, що свідчить про покращення якості оцінок параметрів мережі. У цілому за весь процес навчання логарифм функції правдоподібності збільшився з -4,99 до -2,24.

Наступним прикладом є **мережа CarStarts**, розроблена компанією Norsys, виробником програмного пакету для роботи з мережами Байєса Netica. Топологія мережі зображена на рис. 2.12.

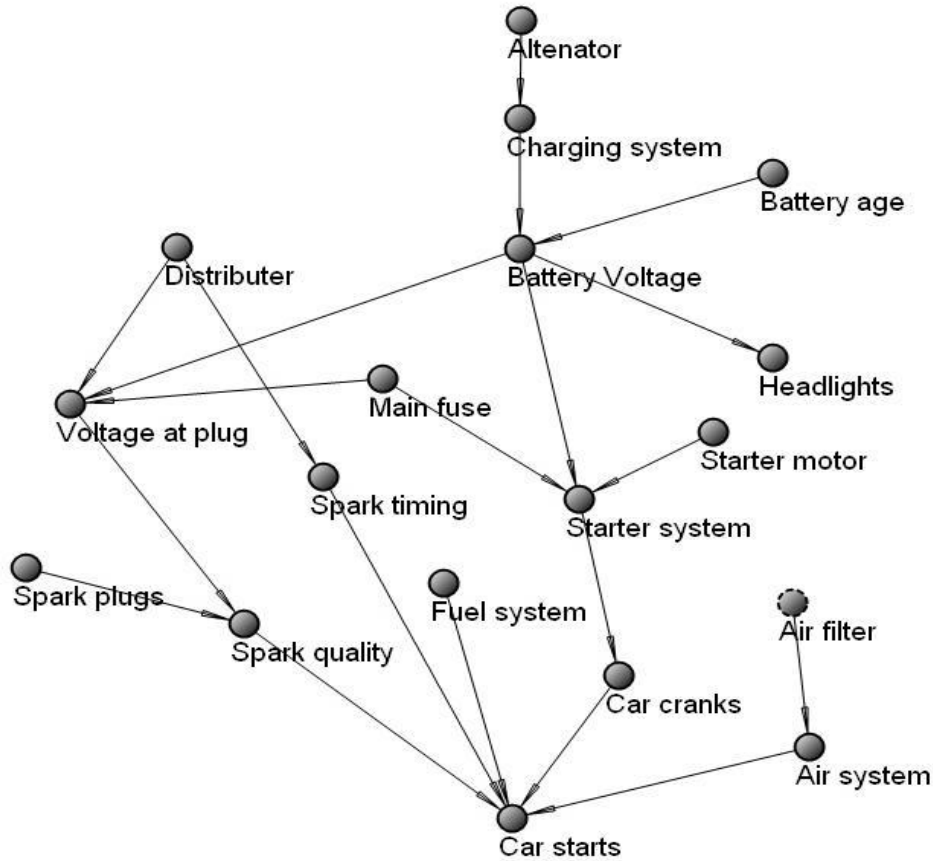


Рис.2.12. Топологія мережі CarStarts

Дана мережа є прикладом байєсівської мережі довіри – одного з найбільш поширених методів представлення знань з невизначеністю. За топологією вона є орієнтованим ациклічним графом. Вузли являють собою випадкові змінні, що можуть бути дискретними чи неперервними. Дуги відображають причинно-наслідкові зв'язки між змінними, "батьківські" вершини відображають причини, а "дочірні" – наслідки. Наприклад, ступінь зарядженості акумулятора впливає на можливість освітлення та роботу системи запалення. Основна задача такої мережі – отримати інформацію про недоступні для діагностування вузли та агрегати автомобіля через дані, що надходять у змінні та спостерігаються і відомі залежності між ними. Дані залежності змодельовані за допомогою відповідних дуг.

Мережа CarStarts містить 18 вершин та 19 дуг. В якості прихованої вершини обрано вершину "Starter system". Кожній змінній мережі ставиться у відповідність таблиця умовних ймовірностей, у якій перераховані всі можливі значення змінної за прийняття її батьківськими вершинами певних значень. У таблиці 2.8 наведено істинні значення ймовірностей станів прихованої вершини та оцінки цих значень, отримані в результаті роботи програми BNetMaster.

Таблиця 2.8

Значення ймовірностей станів прихованої вершини

Стан	Початкові значення	Експериментальні оцінки
Okay	0,596	0,550
Faulty	0,404	0,450

У таблицях 2.9 та 2.10 наведено умовні ймовірності прихованої вершини: відповідно, початкова та експериментально обчислена. Ці значення є дуже близькими, крім випадку Main fuse = "blown" і Starter motor = "Faulty", коли замість однозначного випадку, коли система запуску двигуна не спрацьовує, отримали рівноймовірні події.

Таблиця 2.9

Фактична таблиця умовних ймовірностей прихованої вершини

Main fuse	Starter motor	Battery voltage	Okay	Faulty
Okay	Okay	strong	0,98	0,02
Okay	Okay	weak	0,90	0,10
Okay	Okay	dead	0,10	0,90
Okay	Faulty	strong	0,02	0,98
Okay	Faulty	weak	0,01	0,99
Okay	Faulty	dead	0,01	0,99
Blown	Okay	strong	0,00	1,00
Blown	Okay	weak	0,00	1,00
Blown	Okay	dead	0,00	1,00
Blown	Faulty	strong	0,00	1,00
Blown	Faulty	weak	0,00	1,00
Blown	Faulty	dead	0,00	1,00

Таблиця 2.10

Експериментально обчислена таблиця умовних ймовірностей прихованої вершини

Main fuse	Starter motor	Battery voltage	Okay	Faulty
Okay	Okay	strong	0,88	0,12
Okay	Okay	weak	0,83	0,17
Okay	Okay	dead	0,10	0,90
Okay	Faulty	strong	0,00	1,00
Okay	Faulty	weak	0,00	1,00
Okay	Faulty	dead	0,00	1,00
Blown	Okay	strong	0,00	1,00
Blown	Okay	weak	0,00	1,00
Blown	Okay	dead	0,00	1,00
Blown	Faulty	strong	0,50	0,50
Blown	Faulty	weak	0,50	0,50
Blown	Faulty	dead	0,50	0,50

На рис. 2.13 зображена залежність логарифмічної функції правдоподібності від кількості ітерацій алгоритму ЕМ.

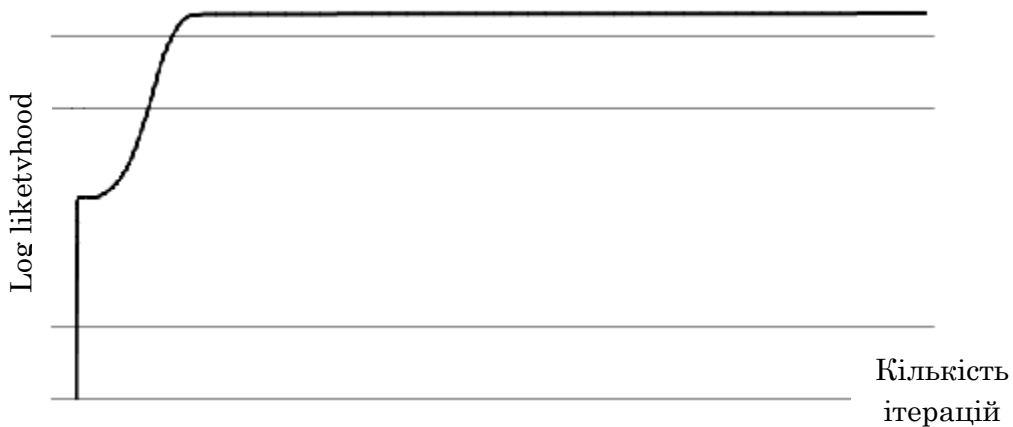


Рис. 2.13. Логарифмічна функція правдоподібності для мережі "Car Starts"

З графіку видно, що з кожною ітерацією значення функції правдоподібності зростає, що свідчить про покращення параметрів мережі. У цілому за весь процес навчання логарифм функції правдоподібності збільшився з -16,90 до -5,97. Але існує певний момент у цьому процесі, коли після деякого числа ітерацій значення функції правдоподібності перестає суттєво змінюватись. Це свідчить про те, що алгоритм має припинити своє функціонування.

### Контрольні питання

1. Наведіть означення системного аналізу. Що є об'єктом вивчення для системного аналізу?
2. Наведіть особливості, ціль та сфери застосування прикладного системного аналізу.
3. Наведіть типи проблем системного аналізу. Як розв'язується проблеми кожного типу?
4. Які переваги надає застосування методології системного аналізу при розв'язанні задач прийняття рішень?
5. Що означає структурованість задачі прийняття рішень?
6. Які задачі прийняття рішень називають слабкоструктурованими?
7. Що виступає інструментом прийняття рішень при застосуванні методології системного аналізу?
8. Які особливості має прикладний системний аналіз?
9. Які існують три класи невизначеностей, пов'язаних із розв'язанням задач системного аналізу?
10. Що означає термін "інтелектуалізація процесу прийняття рішень"?
11. Що таке когнітивна карта і де можна нею скористатись?
12. Назвіть чотири основні характеристики СППР.
13. На яких принципах системного аналізу ґрунтується проектування СППР?
14. Назвіть відомі Вам критерії якості рішень.
15. Що означає ієрархічність досліджуваного об'єкта та його моделі?
16. Що означає навчання мереж Байєса?



17. Наведіть типи навчання мереж Байєса залежно від структури та наявності спостережень. Якими методами вирішуються такі задачі навчання?
18. Що таке приховані вершини в мережі Байєса? Наведіть приклади прихованих мереж. Які переваги використання прихованих вершин при моделюванні?
19. Яким чином можна перетворити приховану вершину у звичайну наявну?
20. Які параметри необхідно оцінювати при навчанні мереж Байєса?
21. Які існують методи навчання параметрів мереж Байєса з прихованими вершинами? Приклади їх використання.
22. Яку роль відіграють приховані змінні при розв'язанні слабкоструктурованих задач?

## РОЗДІЛ 3

### ЙМОВІРНІСНИЙ ВИСНОВОК У БАЙЄСІВСЬКИХ МЕРЕЖАХ

- Поняття ймовірнісного висновку у байєсівській мережі
- Методи оцінювання якості побудови ймовірнісного висновку
- Огляд методів формування ймовірнісного висновку у мережах Байєса
- Формування ймовірнісного висновку у байєсівській мережі на основі навчальних даних

#### 3.1. Поняття ймовірнісного висновку у байєсівській мережі

Побудувавши структуру самостійно або залучивши експертів для її побудови, дослідник отримує мережу Байєса певної топології. Використовуючи її, можна моделювати різноманітні ситуації, тобто задавати вершинам мережі Байєса деякі значення станів і отримувати найбільш ймовірні значення станів для інших вершин мережі.

Процес обчислення оцінки стану вершини на основі апіорної ймовірності про стани інших вершин мережі Байєса називають формуванням (обчисленням, отриманням) *ймовірнісного висновку*. Саме механізм побудови ймовірнісного висновку перетворює будь-яку мережу Байєса, яка описує відповідний процес, на повноцінну складову експертної системи.

Задача побудови ймовірнісного висновку в мережах Байєса є складною з обчислювальної точки зору та неоднозначною. Обсяг обчислень насамперед залежить від кількості вершин та дуг мережі, що з'єднують ці вершини між собою. Неоднозначність задачі полягає в тому, що за різними методами формування ймовірнісного висновку можуть бути одержані різні результати. Така ситуація характерна зокрема для великих мереж Байєса, коли необхідно застосовувати апроксимаційні алгоритми, жертвуючи при цьому точністю обчислень.

Найпростіше ймовірнісний висновок будується на простих деревах у випадках, коли задається стан кореневих вершин. Тоді для обчислення значень станів дочірніх вершин достатньо послідовно застосувати формулу Байєса.

У випадку, коли задається стан дочірньої вершини для обчислення значень станів батьківських вершин, застосовується обернена формула Байєса. В усіх інших випадках, коли структура мережі Байєса є полі-деревом або багатозв'язною мережею, доводиться застосовувати складніші алгоритми.

В якості прикладу побудови ймовірнісного висновку можна навести задачу медичної діагностики, коли за наявності ускладнення дихання у пацієнта, йому потрібно встановити діагноз.

У табл. 3.1 наведено множину навчальних даних  $\langle S, C, B, D \rangle$ , сформовану на підставі 10 спостережень, де  $S$  (smoker) – ймовірність паління;  $C$  (cancer) – вірогідність захворювання на рак;  $B$  (bronchitis) – можливість захворювання бронхітом;  $D$  (dyspnea) – спостереження ознак задишки. При цьому одиниця позначає позитивну відповідь, а нуль – негативну.

Таблиця 3.1

Набір з 10 навчальних спостережень для мережі Байєса

Номер спостереження	Відповіді на навчальні дані			
	S	C	B	D
1	1	1	1	1
2	1	0	1	0
3	1	0	0	0
4	0	0	1	1
5	0	0	0	0
6	1	0	0	0
7	1	1	1	1
8	0	0	0	0
9	0	0	0	1
10	0	0	0	0

Цим даним відповідає структура мережі, зображена на рис. 3.1.

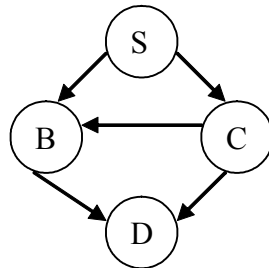


Рис. 3.1 Структура мережі Байєса, що відповідає даним табл. 3.1

Далі, на підставі даних навчальних спостережень, представлених в табл. 3.1, розраховано значення спільного розподілу ймовірностей для всієї мережі (табл. 3.2).

Таблиця 3.2

Значення спільного розподілу ймовірностей всієї мережі

Номер набору даних	Значення вершин				Значення ймовірностей всієї мережі
	S	C	B	D	
1	0	0	0	0	0,3
2	0	0	0	1	0,1
3	0	0	1	1	0,1
4	1	0	0	0	0,2
5	1	0	1	0	0,1
6	1	1	1	1	0,2

Виходячи з даних розрахунків, можна обчислити значення спільного розподілу ймовірностей вершин, які зображені на рис. 3.2.

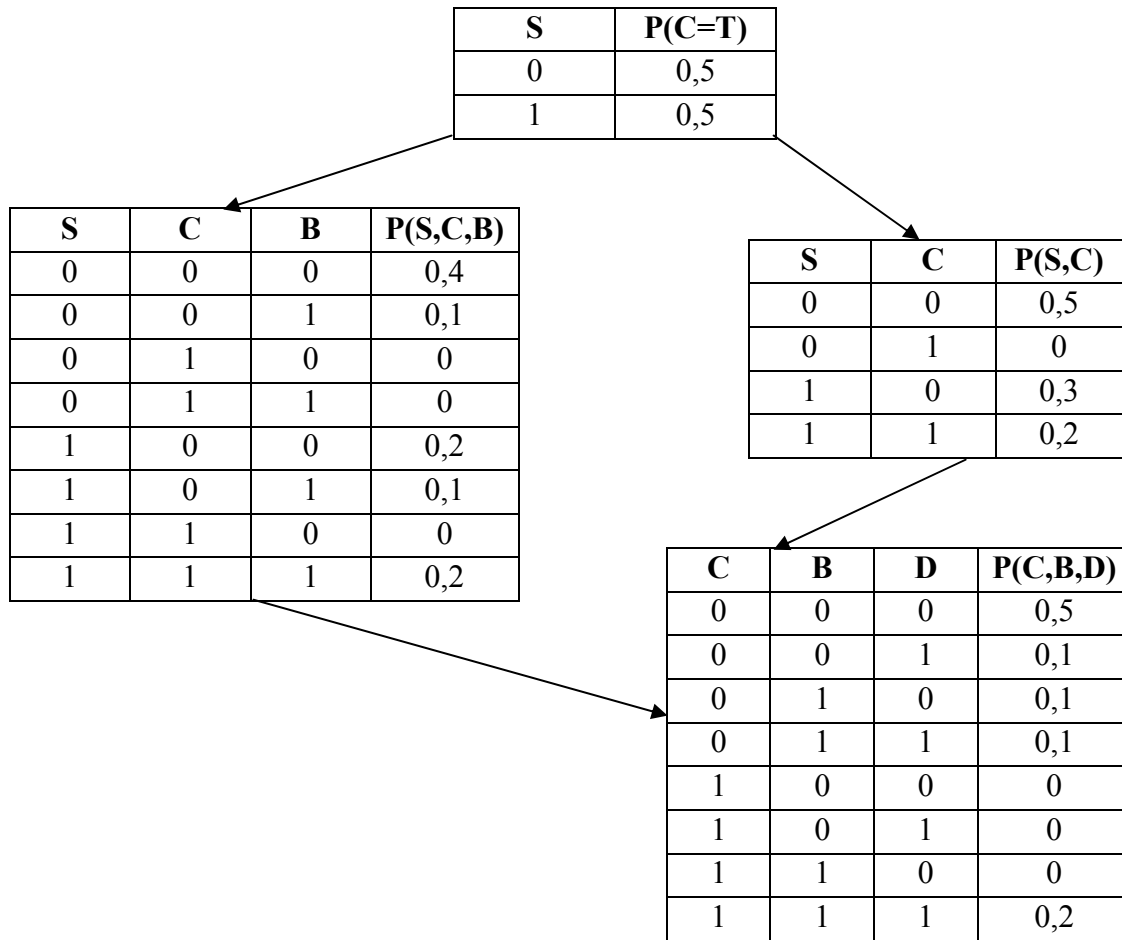


Рис. 3.2. Мережа Байєса у вигляді спільного розподілу ймовірностей вершин, що відповідають даним з табл. 3.1 і рис. 3.1

Використовуючи формулу Байєса  $P(A|B) = \frac{P(A,B)}{P(B)}$  за значеннями спільного розподілу ймовірностей вершин і значеннями ймовірностей вершин, можна розрахувати значення умовних ймовірностей вершин, які, у свою чергу утворюють таблиці умовних ймовірностей.

Повна спільна ймовірність розглянутої мережі Байєса обчислюється за формулою:

$$P(S,C,B,D) = P(S) \cdot P(C|S) \cdot P(B|S,C) \cdot P(D|B,C) \tag{3.1}$$

Розраховані значення представлено в табл. 3.3.

Емпіричні значення спільного розподілу ймовірностей всієї мережі і розраховані на основі таблиці умовних ймовірностей відрізняються між собою – це можна побачити з табл. 3.2 і рис. 3.3.

Таблиця 3.3

Значення спільного розподілу ймовірностей для всієї мережі

Номер набору даних	Значення вершин				Значення ймовірностей всієї мережі
	S	C	B	D	
1	0	0	0	0	0,332
2	0	0	0	1	0,068
3	0	0	1	0	0,05
4	0	0	1	1	0,05
5	0	1	0	0	0
6	0	1	0	1	0
7	0	1	1	0	0
8	0	1	1	1	0
9	1	0	0	0	0,164
10	1	0	0	1	0,034
11	1	0	1	0	0,051
12	1	0	1	1	0,051
13	1	1	0	0	0
14	1	1	0	1	0
15	1	1	1	0	0
16	1	1	1	1	0,2

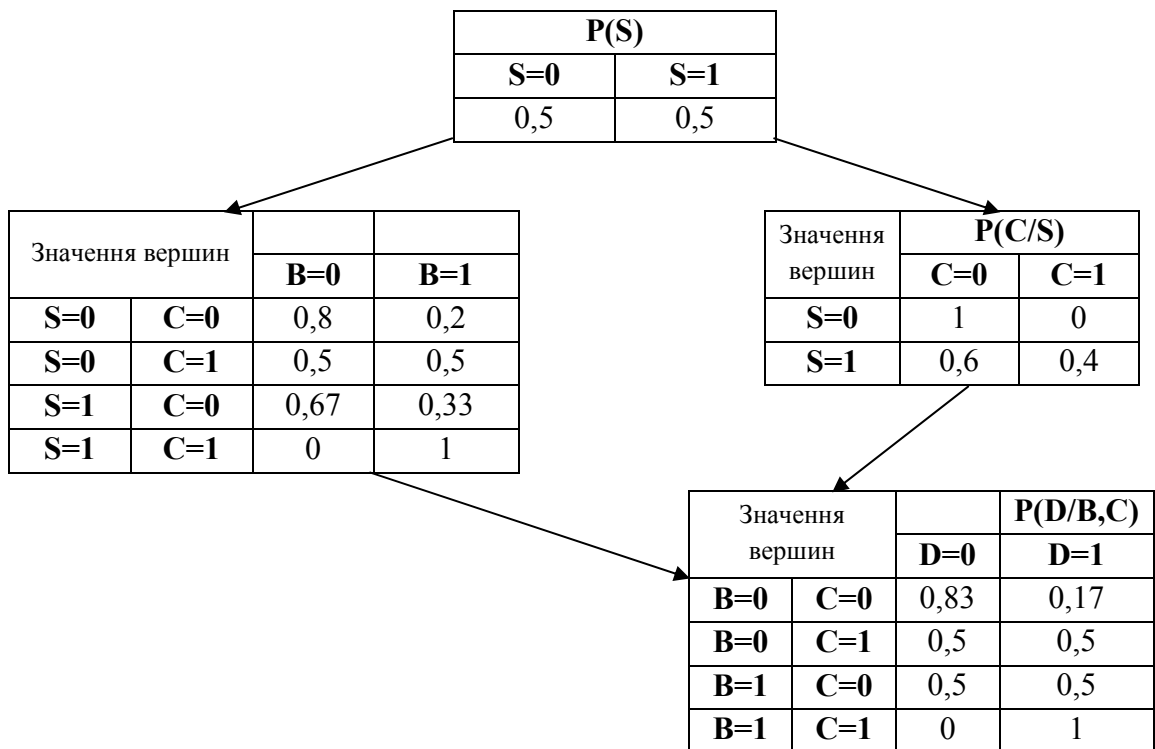


Рис. 3.3. Мережа Байєса у вигляді таблиці умовних ймовірностей вершин, що відповідають даним табл. 3.1

Зазначимо, що не тільки за числовими значеннями, але в результаті використання таблиці умовних ймовірностей, з'являються нові події, яких не було серед навчальних даних. Це пов'язано з частковою втратою інформації при переході від емпіричних навчальних даних до таблиці умовних ймовірностей.

Отже, метою ймовірнісного висновку є знаходження  $P(A|B)$  апостеріорної ймовірності шуканих вершин  $A$ , при деякому значенні спостережуваних вершин  $B$ .

Задача побудови ймовірнісного висновку є складною з обчислювальної точки зору та неоднозначною, тобто неможливо знайти єдиний алгоритм побудови, який демонстрував би найкращі результати для всіх типів мереж.

За розміром вирішуваних задач методи формування ймовірнісного висновку в мережах Байєса можна розділити на дві групи: методи точного висновку та апроксимаційні [57, 58].

У табл. 3.4 наведено перелік методів, з яких складається кожна з груп.

Таблиця 3.4

Класифікація методів обчислення ймовірнісного висновку в мережах Байєса

Алгоритми точного висновку	Апроксимаційні алгоритми
1. Алгоритм Перла розповсюдження повідомлення для однозв'язних мереж (полі-дерев)	1. Часткового або неповного висновку (exact inference partially)
2. Кластеризації дерева клік (clique tree clustering)	2. Варіаційні методи (variational algorithms) використовуються для обчислення середніх ознак великих мереж
3. Визначаючого перетину (cutest conditioning)	
4. Виключення змінних (variable elimination algorithm)	3. Методи стохастичного вибору (stochastic sampling)
5. Символьного ймовірнісного висновку (symbolic probabilistic inference)	4. Пошукові методи (search-based), основані на евристичних алгоритмах пошуку, які використовують при переході від задачі обчислення ймовірнісного висновку до оптимізаційної задачі
6. Диференціальний підхід (differential method)	

3.2. Алгоритми формування точного висновку

*Алгоритм Перла для розповсюдження повідомлення в однозв'язних мережах*

У 1986 році Джуді Перл [59] запропонував для обчислення значень ймовірностей в мережах довіри ідею використання інформаційних повідомлень, спрямованих від дочірніх та батьківських вершин. У своїй наступній роботі 1988 року він [24] представив алгоритм розповсюдження повідомлення, інша назва якого "ітераційний алгоритм розповсюдження довіри (IBP – iterative belief propagation) для

полі-дерев". Метою цього алгоритму є "... проникнення та розповсюдження впливу нових подій або даних байєсівської мережі таким чином, щоб в результаті кожна пропозиція була безсумнівно повністю узгодженою з аксіоматикою теорії ймовірностей" [24, с. 143]. Процес розповсюдження ймовірностей у мережах Байєса є основним процесом, на якому ґрунтується висновок, тобто прийняття рішення: "Кожна нова подія або дані розглядаються як збурення, яке розповсюджується по мережі завдяки повідомленням, якими обмінюються сусідні вершини" [24, с. 143].

Ідея алгоритму полягає в тому, що при обчисленні умовної ймовірності (міри довіри) вершини  $X$ , використовується інформація про стан вершин-сусідів. На рис. 3.4 показано, як інформація про стан батьківської вершини передається через  $\pi$ -повідомлення, а через  $\lambda$ -повідомлення передається інформація про дочірню вершину.

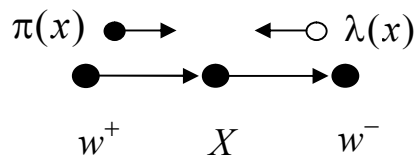


Рис. 3.4. Вершина  $X$  отримує повідомлення від дочірньої та батьківської вершин-сусідів

У випадку мережі Байєса, зображеної на рис. 3.4, ймовірність (значення довіри) для вершини  $X$  обчислюється за формулою:

$$P(x) = P(x | w) = P(x | w^+, w^-) = \frac{P(x | w^+) \cdot P(w^- | x)}{P(w^+, w^-)} = \alpha \cdot P(x | w^+) \cdot P(w^- | x) = \alpha \cdot \pi(x) \cdot \lambda(x) \quad (3.2)$$

У випадку мережі з однією батьківською вершиною та однією дочірньою вершиною застосовується формула (3.3):

$$P(x) = \alpha \cdot \pi(x) \cdot \lambda(x), \quad (3.3)$$

де  $\alpha$  – нормуюча константа.

Нехай  $W$  - множина, яка містить всі змінні, для яких можна отримати поточні значення (спостереження). Таку множину називають інстанційованою. Множина інстанційованих змінних складається з двох підмножин:  $W_X^-$  - множина, яка містить всі інстанційовані змінні, що знаходяться нижче вузла  $X$ ;  $W_X^+$  - множина, яка містить всі інстанційовані змінні, що знаходяться над вузлом  $X$ . Для спрощення записів застосовують такі позначення:  $W_B^- = \lambda$  і  $W_X^+ = \pi$ .

До деякої довільної вершини  $X$  мережі інформація може надходити від батьківських вузлів або від вузлів-нащадків. Так,  $\pi(x)$  представляє інформацію, яка надходить від батьківської вершини  $X$ .  $\pi(x)$  називають також *прямим оператором* (або *форвард-оператором*), оскільки він відображає розповсюдження інформації в

напрямку до листків мережі (зверху-вниз). У випадку, коли конкретна змінна отримує нову інформацію від батьківських вершин, вона повинна бути передана далі (вперед) до дочірніх вершин, якщо вони є. Розповсюдження інформації в напрямку “зверху-вниз” (тобто в напрямку “до листків”) можна концептуалізувати за допомогою  $\pi$  - повідомлення. Значення  $\pi$  – це внутрішнє значення однієї змінної, яке використовується з метою обчислення величини довіри (умовної ймовірності) для однієї змінної.

Подібно до  $\pi$ , значення  $\lambda$  також необхідно розглядати як  $\lambda$  - повідомлення і  $\lambda$  - значення.  $\lambda$  - повідомлення означає, що інформація передається “знизу-вверх”, тобто – від листків до кореня, а  $\lambda$  - значення – це внутрішнє значення (величина) змінної.  $\lambda$  називають *зворотним оператором*, оскільки він свідчить про розповсюдження інформації в напрямку до кореня; він також представляє собою ретроспективне (діагностичне) підтвердження того, що від нащадків вузла  $X$  отримано інформацію, що  $X = x$ . Використання означення “діагностичний” є виправданим, оскільки  $\lambda$  - повідомлення поступають у мережу тоді, коли інстанціюється (приймає конкретне значення) один або більше атрибутів.  $\lambda$  - повідомлення безпосередньо впливає на кожне  $\lambda$  - значення змінної [60].

Наприклад, нехай для мережі, наведеної на рис. 3.5, надходять спостереження щодо змінних  $B, D, F, H$ , тобто  $W = [B, D, F, H]$ . Тоді для вузла  $C$  підмножина  $W_C^- = [F, H]$  і  $W_C^+ = [B, D]$ .

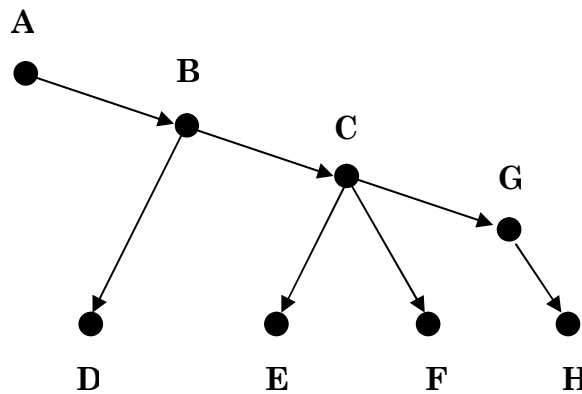


Рис.3.5. Приклад формування інстанційованих множин

Таким чином, значення  $\lambda$  будуть поступати від дочірніх вершин, що можна формально представити у вигляді:

$$\lambda(b_i) = \begin{cases} p(W_B^- | b_i), & \text{якщо } B \notin W \\ 1, & \text{якщо } B \in W \text{ і } b_i \text{ – інстанційоване значення,} \\ 0, & \text{якщо } B \in W \text{ і } b_i \text{ – неінстанційоване значення,} \end{cases} \quad (3.4)$$

де  $b_i$  конкретне значення із множини  $b$ , які може приймати змінна (вузол)  $B$ , тобто  $b_i$  формують множину  $b$ .



Іншими словами, стосовно виразу (3.8) можна сказати: якщо  $B \in W$  і  $b_i$  вже отримало значення від датчика чи експерта, то  $\lambda(b_i) = 1$ , а якщо  $B \in W$ , але  $b_i$  ще не отримало значення, то  $\lambda(b_i) = 0$ .

Значення для  $\pi$  формалізується простіше:

$$\pi(b_i) = p(b_i | W_B^+). \quad (3.5)$$

Обоє  $\pi$  і  $\lambda$  - значення використовуються для оновлення величини довіри змінної.

*Довіра* – це розподіл ймовірностей, який ґрунтується на всій інформації, що міститься в мережі. Таким чином, можна виразити загальний ступінь довіри до виразу " $X = x$ " шляхом об'єднання і нормування  $\pi$  і  $\lambda$  - значень:

$$Bel(x) = \alpha \lambda(x) \pi(x), \quad (3.6)$$

де  $\alpha$  - нормуюча константа.

Рівняння (3.10) характеризує розподіл ймовірностей деякого атрибута. Якщо довіра будь-якого вузла змінюється, він посилає  $\pi$  - повідомлення до своїх нащадків. Тобто  $\pi$  - повідомлення завжди передаються в прямому напрямку по мережі, коли з'являється спостереження однієї змінної. Тому  $\pi$  - повідомлення і  $\pi$  - значення називають "діагностичною" інформацією.

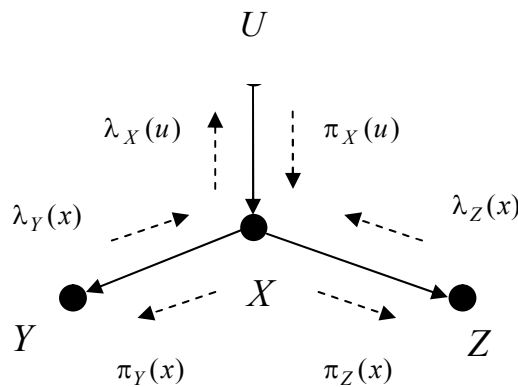


Рис. 3.6. Приклад мережі Байєса, де вершина  $X$  одержує повідомлення від двох дочірніх вершин та однієї батьківської

У випадку, зображеному на рис. 3.6, коли потрібно обчислити значення довіри вершини з однією батьківською вершиною і двома дочірніми, застосовується формула 3.7:

$$Bel(x) = \alpha \lambda_Y(x) \lambda_Z(x) \sum_u M_{x|u} \pi_X(u), \quad (3.7)$$

де  $M_{x|u}$  - матриця, в якій зберігаються всі умовні ймовірності між змінними  $U$  і  $X$ .

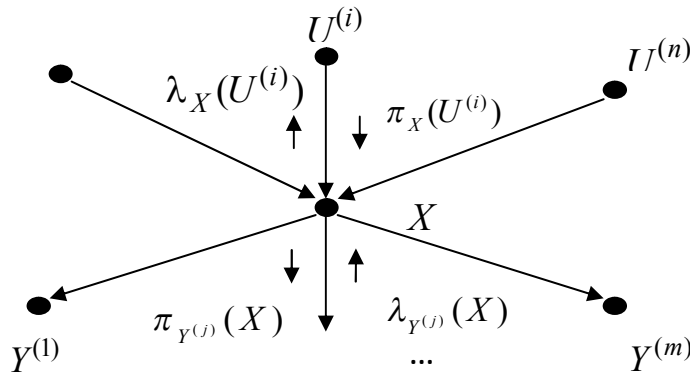


Рис. 3.7. Загальний випадок: вершина  $X$  має  $n$  батьківських та  $m$  дочірніх вершин

У загальному випадку, зображено  $Y^{(j)}$  му на рис. 3.7, коли вершина має  $m$  дочірніх вершин та  $n$  батьківських вершин, обчислення виконується за формулою (3.11), але з урахуванням формул (3.8) та (3.9).

$$\lambda(x) = \begin{cases} \prod_{j=1}^m \lambda_{Y^{(j)}}(x), & \text{якщо } B \notin W, \\ 1, & \text{якщо } B \in W \text{ і } b_i - \text{інстанцією ване значення,} \\ 0, & \text{якщо } B \in W \text{ і } b_i - \text{неінстанцією ване значення} \end{cases} \quad (3.8)$$

$$\pi(x) = \sum_{c_1=1}^{k_1} \dots \sum_{c_n=1}^{k_n} \left( p(x | u_{c_1}^{(1)}, \dots, u_{c_n}^{(n)}) \cdot \pi_x(u_{c_1}^{(1)}) \dots \pi_x(u_{c_n}^{(n)}) \right) \quad (3.9)$$

При цьому вважається, що кожна  $i$ -та батьківська вершина може приймати  $k_i$  станів  $i=1 \dots n$ , а кожна  $j$ -та дочірня вершина  $r_j$  станів:  $r=1 \dots m$ . Тобто значення  $\lambda$  конкретної змінної є добутком повідомлень, які надходять до неї від її дочірніх вершин, а на  $\pi$  - повідомлення впливають всі  $\pi$  - повідомлення, які приймає одна змінна. Після перемноження комбіноване  $\pi$  - повідомлення надсилається дочірнім вершинам. Процес розповсюдження інформації ілюструє рис.3.8 на прикладі двох інстанційованих змінних.

На рисунку 3.8 послідовно представлено виконання кроків алгоритму Перла:

Рис. 3.8а – Ініціалізація дерева.

Рис. 3.8б – Інстанціювання двох змінних;  $\lambda$  - повідомлення (стрілочки без заповнення) починають розповсюджуватись в напрямку кореневої змінної.

Рис. 3.8в –  $\lambda$  - повідомлення продовжують розповсюджуватись у напрямку кореневої змінної. Дві змінні, які отримали  $\lambda$  - повідомлення (з рис.3.8б), змінили рівні своєї довіри, а тому посилають свої  $\pi$  - повідомлення (стрілочки із заповненням) у напрямку змінних, представлених листками.

Рис. 3.8г –  $\lambda$  - повідомлення досягли кореневої змінної, для якої оновлюється рівень довіри. Відразу ж після оновлення коренева змінна посилає нові  $\pi$  - повідомлення в напрямку змінних-листіків. Разом з тим,  $\pi$  - повідомлення (з рис.3.8в) приходять до останніх змінних-листіків.

Рис. 3.8д –  $\pi$  - повідомлення кореневої змінної досягають змінних-листіків.

Рис. 3.8е – Ситуація така ж, як і на рис. 3.8д:  $\pi$  - повідомлення досягають змінних-листіків, і мережа переходить у стан рівноваги.

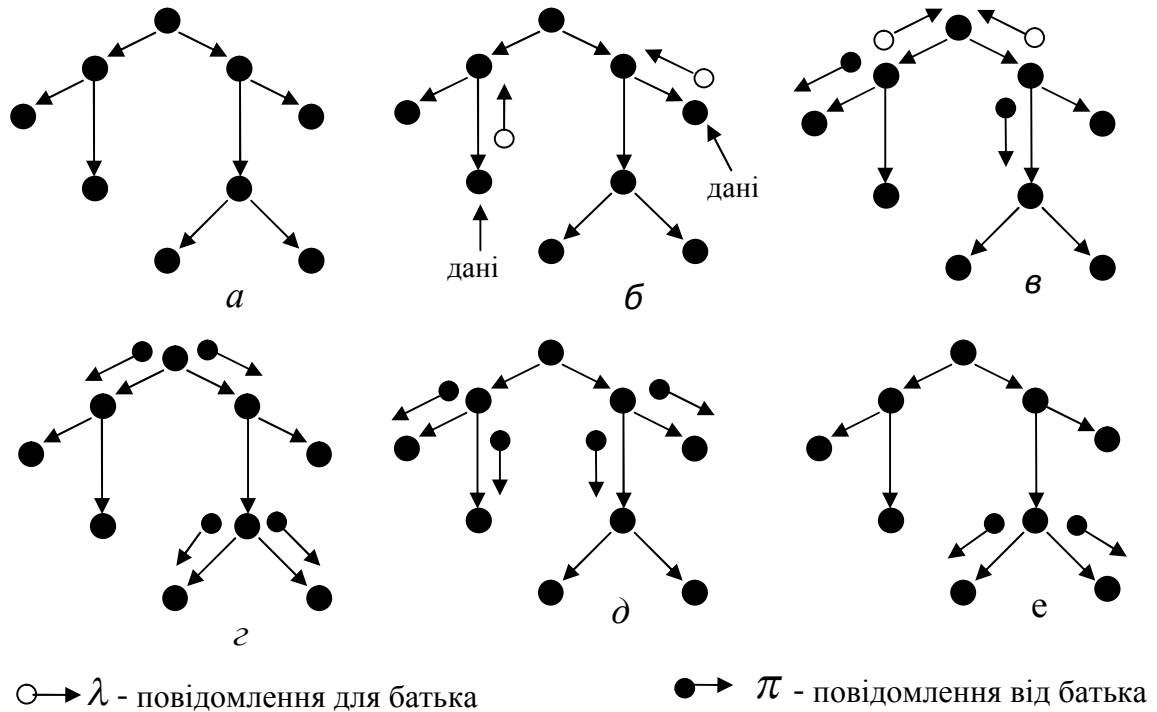


Рис. 3.8. Схематичне представлення розповсюдження інформації по полі-дереву в алгоритмі Перла

Складність цього алгоритму залежить від кількості вершин у дереві.

### Алгоритми кластеризації

У 1988 році С. Лорітценом та Д. Шпігельхалтером (S. L. Lauritzen and D. J. Spiegelhalter) [56, 57, 61, 62] запропоновано **LS-метод**.

Пізніше, у 1990 році, Ф. Дженсенем, К. Олесеном, С. Андерсенем [43] було запропоновано модифікований LS-метод, який зустрічається в літературі під назвою **Hugin**, яку він отримав завдяки використанню у програмному комплексі Hugin Expert [43], розробленому Ф. Дженсенем та його групою.

У 1997 році С. Лорітцен і Ф. Дженсен [64] розвинули Hugin-метод настільки, що його стало можливо застосовувати в інших теоріях, у тому числі в теорії функцій довіри Демпстера-Шафера (Dempster-Shafer).

Під впливом робіт С.Лорітцена, Д. Шпігельхалтера [56] та Дж. Перла [59], у 1997 році П. Шеной (P. Shenoy) розробив новий метод, який отримав назву **SS-метод** (Shenoy-Shafer) [65, 66]. Як показали проведені експерименти [67], з обчислювальної точки зору SS-метод ефективніший у порівнянні з Hugin та LS-методами, а Hugin кращий за LS-метод. Але за критерієм обсягу пам'яті, необхідної для зберігання проміжної обчислювальної інформації, найкращим виявився LS-метод, а найгіршим SS.

Пізніше, у 1998 році, А. Мадсенем та Ф. Дженсоном запропоновано **алгоритм лінивого розповсюдження** (lazy propagation) [68], який є комбінацією SS-алгоритму та алгоритму поглинаючого виключення [69, 70]. Крім цієї комбінації, існує ще декілька модифікацій лінивого алгоритму, зокрема, його комбінації з SS та SPI [71-74], а також з SS та AR [75] алгоритмами.

Ідея *LS-методу ймовірнісного висновку* є основоположною для методів кластеризації, вона полягає в тому, що для реалізації ймовірнісного висновку необхідно спочатку привести структуру мережі Байєса до вигляду об'єданого дерева, а потім використовувати алгоритм розповсюдження повідомлень по дереву догори та донизу і послідовно перераховувати таблиці умовних ймовірностей вершин дерева. У літературі, замість терміну “об'єдані дерева” (junction trees), іноді застосовуються терміни *дерева клік* (clique trees), *гіпердерева* (hypertrees) та *якісні дерева Маркова* (qualitative Markov trees).

Загалом використання об'єданих дерев дозволяє застосовувати ідею Дж. Перла обчислення ймовірнісного стану за допомогою  $\lambda$  і  $\pi$  - повідомлень. Серед подібних методів можна виділити *GDL-метод* (generalized distributive law), який фактично є симбіозом цілої низки методів. Його запропоновано С. Аджи та Р. Мак Елісом у 2000 році [76]. Суть методу полягає в передачі повідомлень по об'єданому дереву із застосуванням локальних ядер (local kernel) вузлів об'єданого дерева.

*LS-алгоритм* [56, 57, 320,77, 78] реалізується у два етапи.

На **першому етапі** виконується побудова об'єданого дерева клік з первинної структури мережі та заповнення вершин цього дерева таблицями умовних ймовірностей мережі. Даний етап реалізується у чотири кроки:

- Крок – 1. Моралізування графа.
- Крок – 2. Триангуляція графа.
- Крок – 3. Ідентифікація клік.
- Крок – 4. Побудова об'єданого дерева.

**Другий етап** передбачає обчислення значень ймовірностей станів вершин на основі алгоритмів розповсюдження значень ймовірності по об'єданому дереву. Для обчислення значень ймовірностей клік використовуються  $\lambda$  і  $\pi$  - повідомлення. Після цього за значеннями ймовірностей клік обчислюють індивідуальні ймовірності вершин. Обчислювальна складність LS алгоритму дорівнює  $O(p \cdot r^m)$ , де  $p$  - кількість вершин мережі, а  $r$  - максимальна кількість станів, які може набувати вершина.

**Перший етап: побудова об'єданого дерева клік** передбачає виконання серії перетворень ациклічного спрямованого графа  $G$  (рис. 3.9 а) для його приведення до вигляду об'єданого дерева.

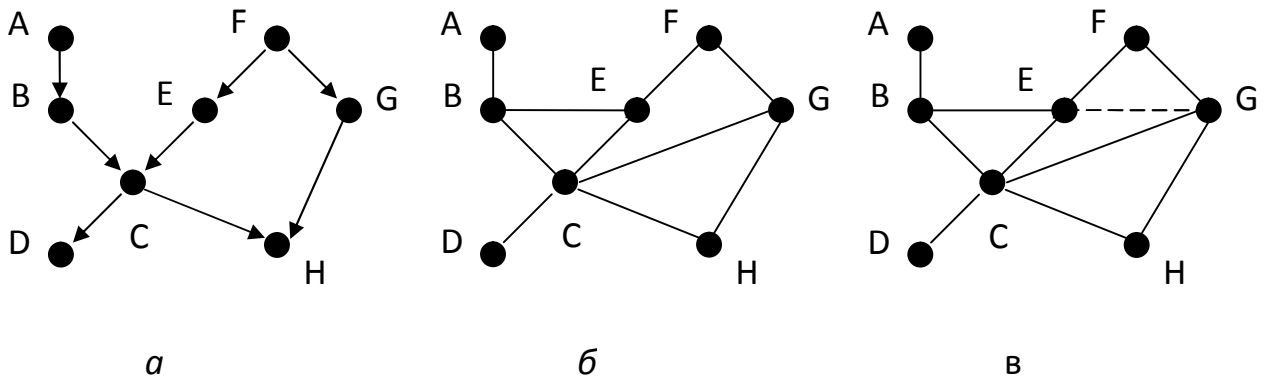


Рис. 3.9. а) Ациклічний спрямований граф  $G$ ; б) моралізований граф  $G_m$ ; в) триангульований (розбитий на трикутники) граф  $G_u$

**Крок – 1. Моралізація графа  $G$**  (рис. 3.9 а) до вигляду графа  $G_m$  (рис. 3.9 б): послідовно перебираються всі вершини мережі Байєса, у яких є батьки, якщо батьки вершини не зв'язані між собою – з'єднати дугами взаємно незв'язані вершини-батьки кожної вершини мережі. Якщо батьки вершини не зв'язані між собою, то між ними вводиться зв'язок ("сусід"). Таке перетворення називають ще "одруженням" вершин-батьків, після чого слід дезорієнтувати граф – спрямовані дуги графа замінити неспрямованими (для всіх вершин "батько-нащадок" замінити на зв'язок "сусід").

**Крок – 2. Триангулювати моралізований граф  $G_m$**  (рис. 3.9 б) до вигляду графа  $G_u$  (рис. 3.9 в), тобто представити його так, щоб довжина циклів не перевищувала трьох. Графічно ця операція означає розбиття моралізованого графа на трикутники. Для триангуляції застосовують алгоритм часової заміни [79, 80] (fill-in computation algorithm). Також на цьому кроці застосовують алгоритм пошуку максимальної потужності [81-83] (maximum cardinality search) для побудови упорядкованої множини вершин, яка позначається як  $\alpha$ . Процес триангуляції можна описати таким чином:

1. Послідовно перебирати всі вершини мережі Байєса, доки не будуть розглянуті всі вершини.

Для цього слід виконати наступні дії:

- перевірити, чи є суміжними між собою "сусіди" аналізованої вершини, якщо так, то така вершина симпліціальна – утворює кліку разом із своїми сусідами (така вершина виключається з розгляду разом із її ребрами);

- якщо після перебору всіх вершин мережа, що залишилася до розгляду, не порожня, то перейти до наступного кроку алгоритму; у протилежному випадку, вважається, що граф триангульований;

2. У мережі, що залишилась до розгляду, послідовно перебираються вершини: шукається вершина з найбільшою кількістю сусідів (така вершина стає симпліціальною шляхом введення додаткових ребер між її несуміжними сусідами), тобто тепер вона утворює кліку з сусідами, а потім ця вершина виключається із розгляду. Після розгляду усієї мережі до початкового моралізованого ненаправленого графа додаються додаткові ребра, і такий граф буде триангульованим.

**Крок – 3.** За допомогою алгоритму пошуку клік [84, 85] (cliques-finding algorithm) в триангульованому моралізованому графі  $G_u$  (рис. 3.9 в) **визначається множина клік** (рис. 3.10), тобто підграфів, потужності яких не перевищують трьох.

Для поточної кліки слід виконати такі дії: перевірити, чи є вона підмножиною інших не перебраних клік, якщо так, то вона знищується; якщо в ній та інших кліках співпадає хоча б одна вершина, то між відповідними кліками вводиться ребро, що містить сепаратор – перетин множин вершин цих клік. Після цього виконується ранжирування множини клік, застосовуючи упорядковану множину вершин  $\alpha$ . Ранжирування множини клік робиться таким чином: першою клікою стає та, що має вершину, яка в упорядкованій множині вершин  $\alpha$  посідає перше місце; якщо клік, які мають цю вершину є декілька, то дивляться на наявність у кліці вершини, яка займає друге місце в упорядкованій множині вершин  $\alpha$ , і так далі. Таким чином виконується ранжирування усієї множини клік.

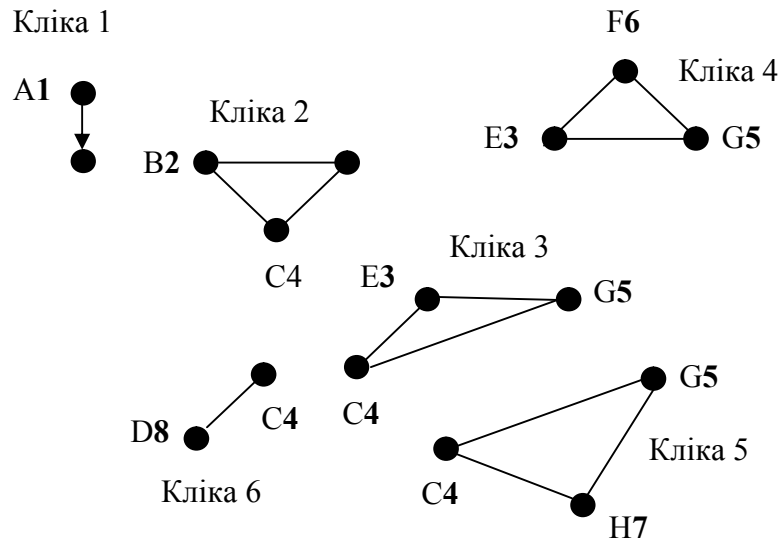


Рис. 3.10. Ранжирувана множина клік триангульованого графа  $G_u$

**Крок – 4. Побудова об'єданого дерева** [61, 78]. Під час реалізації даного кроку всі кліки послідовно зв'язуються між собою шляхом вибору для зв'язку ребер з найбільшими сепараторами, а інші ребра видаляються. Дерево, що зв'язує всі кліки з максимальними потужностями сепараторів, буде об'єднаним деревом. Для зручності, як корінь в об'єданому дереві, вибирається кліка з найбільшою кількістю вершин. Якщо таких вершин декілька, то перевага віддається кліці з найбільшою кількістю ребер.

Вхідними даними є ранжирувана множина клік триангульованого графа  $G_u$ . Алгоритм побудови об'єданого дерева можна представити наступним чином:

1.  $S_i = \text{Кліка}_i \cap (\text{Кліка}_1 \cup \text{Кліка}_2 \cup \dots \cup \text{Кліка}_{i-1})$
2.  $R_i = \text{Кліка}_i - S_i$ , тобто  $R_i = \text{Кліка}_i \cap S_i$ .
3. Якщо  $S_i \subseteq \text{Кліка}_j$  та  $j < i$  то  $\text{Кліка}_j$  є батьківською для  $\text{Кліки}_i$ .
4. Обчислення потенційної ймовірності  $i$ -ї кліки за формулою:

$$\psi_i = \psi(\text{Кліка}_i) = \prod_{\forall v \in \text{Кліка}_i} P(v | pa(v)), \text{ де запис } \forall v \in \text{Кліка}_i \text{ означає перебір всіх}$$

вершин  $v$   $i$ -ї кліки, а  $pa(v)$  – множина батьків вершини  $v$ , які разом із  $v$  знаходяться в  $i$ -й кліці. У випадку, коли батьки є, але в  $i$ -й кліці знаходяться не всі батьки, або їх немає зовсім в цій кліці, то  $P(v | pa(v)) = 1$ . Значення ймовірностей для розрахунків беруться з оригінальної мережі Байєса (рис. 3.9 а).

5. Для кожної вершини об'єданого дерева (кліки) обчислюються та зберігаються параметри  $\text{Кліка}_i, S_i, R_i, \psi(\text{Кліка}_i)$ .

**У результаті** буде побудоване дерево клік, тобто об'єдане дерево (рис. 3.11), побудоване за мережею Байєса, яке представлено на рис. 3.9а.

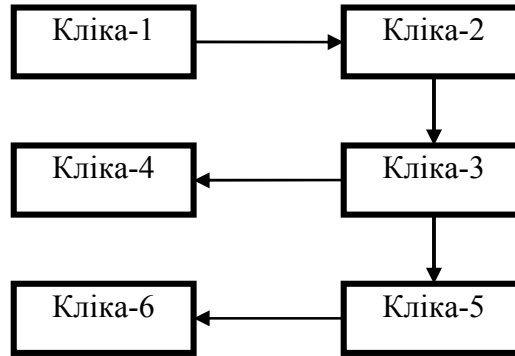


Рис. 3.11 Об'єднане дерево (дерево клік)

На завершення першого етапу об'єднане дерево заповнюється таблицями. Заповнення відбувається поетапно, в ході обходу вершин формується таблиця значень для об'єданого дерева (табл. 3.5).

Таблиця 3.5

Таблиця значень для об'єданого дерева, представленого на рис. 3.11

Номер кліки	Вершини кліки $V$	Граничні вершини $S_i$	Остаточні вершини $R_i$	Функція потенційної ймовірності кліки $\psi_i = \prod_{\forall v \in \text{Кліка}_i} P(v   pa(v))$
1	$A, B$	-	$A, B$	$\psi_1 = P(A   pa(A)) \cdot P(B   pa(B)) = P(A) \cdot P(B   A)$
2	$B, C, E$	$B$	$C, E$	$\psi_2 = P(B   pa(B)) \cdot P(C   pa(C)) \cdot P(E   pa(E)) = P(B   \{\text{не батьки}\}) \cdot P(C   B, E) \cdot P(E   \{\text{не всі батьки}\}) = 1 \cdot P(C   B, E) \cdot 1 = P(C   B, E)$
3	$C, E, G$	$C, E$	$G$	$\psi_3 = P(C   pa(C)) \cdot P(E   pa(E)) \cdot P(G   pa(G)) = P(C   \{\text{не всі батьки}\}) \cdot P(E   \{\text{не всі батьки}\}) \cdot P(G   \{\text{не всі батьки}\}) = 1 \cdot 1 \cdot 1 = 1$
4	$E, F, G$	$E, G$	$F$	$\psi_4 = P(E   pa(E)) \cdot P(F   pa(F)) = P(G   pa(G)) = P(E   F) \cdot P(F) \cdot P(G   F)$
5	$C, G, H$	$C, G$	$H$	$\psi_5 = P(C   pa(C)) \cdot P(G   pa(G)) \cdot P(H   pa(H)) = P(C   \{\text{не всі батьки}\}) \cdot P(G   \{\text{не всі батьки}\}) \cdot P(H   C, G) = 1 \cdot 1 \cdot P(H   C, G) = P(H   C, G)$
6	$C, D$	$C$	$D$	$\psi_6 = P(C   pa(C)) \cdot P(D   pa(D)) = P(C   \{\text{не всі батьки}\}) \cdot P(D   C) = 1 \cdot P(D   C) = P(D   C)$

$S_i$  - граничні вершини  $i$  - ї кліки (вершини, що належать батьківським клікам).

$R_i$  – остаточні вершини  $i$  - ї кліки (вершини, які не належать батьківським клікам).

$\psi(\text{Кліка}_i)$  – значення ймовірності  $i$  - ї кліки.

Заповнення починається з листів дерева і послідовно перебираються усі кліки. Розглядаються "невідмічені" вершини в необробленій кліці:

1. Якщо серед "невідмічених" вершин є така, що не зустрічається в інших необроблених кліках, то таблиця повинна мати вигляд  $P(\text{this node} | \text{other nodes in clique})$  у такому випадку ця вершина "відмічена" і кліка вважається обробленою.

2. Якщо в первинній мережі Байєса такої таблиці немає або така вершина не одна, то використовується таблиця сукупного розподілу ймовірностей, у такому разі усі вершини кліки "відмічені", і кліка оброблена.

3. Якщо в кліці вже є "відмічені" вершини і всього одна "невідмічена" вершина, то використовується таблиця виду  $P(\text{unmarked node})$  з первинної мережі Байєса - у такому випадку ця вершина "відмічена", і кліка вважається обробленою.

4. Якщо в кліці вже є "відмічені" вершини, а "невідмічених" вершин декілька, то використовується таблиця сукупного розподілу ймовірності "невідмічених" вершин, і тоді ці вершини "відмічені", а кліка вважається обробленою.

5. Якщо в необробленій кліці усі вершини вже "відмічені", то кліка заповнюється таблицею виду  $P(\text{node any})$  і вважається обробленою.

За отриманими значеннями ймовірностей клік, можна обчислити значення спільної ймовірності об'єднаного дерева:

$$P(\text{Мережі}) = P(A, B, C, D, E, F, G, H) = P(\text{Кліка}_1, \dots, \text{Кліка}_6) = \prod_i \text{Кліка}_i = \prod_i \psi_i \quad (3.10)$$

**Другий етап – алгоритм пропagaції:** формування ймовірнісного висновку за об'єднаним деревом. Для кожного спостереження змінної вибирається одна таблиця, яка містить цю змінну. Задаємо нульовими усі входження, які суперечать спостереженню. В результаті вибирається одна таблиця, яка містить цю змінну. Задаємо нульовими усі входження, які суперечать спостереженню.

Ймовірнісний висновок за об'єднаним деревом за своєю ідеєю співпадає з алгоритмом Перла.

На рис. 3.12 представлено процес розповсюдження повідомлень по об'єднаному дереву.

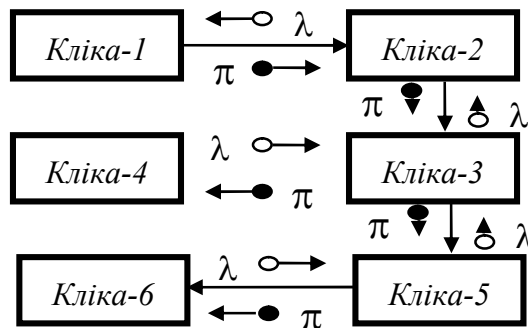


Рис. 3.12. Схематичне представлення розповсюдження інформації у вигляді  $\lambda$  і  $\pi$  – повідомлень за вузлами об'єднаного дерева (дерева клік)



Спочатку виконують *процедуру "сходження догори"* (upword). Повідомлення є результатом маргіналізації - підсумовування змінних таблиць, які не містяться в сепараторі. Після відсилання повідомлення відправник ділить свою поточну таблицю умовних ймовірностей на нього. Коли отримувачу надходить повідомлення, він множить його на свою таблицю умовних ймовірностей і виходить нова таблиця. Коли він отримує повідомлення від усіх своїх нащадків, то, у свою чергу, посилає також повідомлення своєму батьку і ділить свою таблицю на відіслане повідомлення. Процес триває доти, доки корінь зв'язного дерева не отримає повідомлення від усіх своїх нащадків.

Потім виконують *процедуру "сходження донизу"* (downward). Корінь посилає повідомлення кожному своєму нащадкові. Він ділить свою таблицю умовних ймовірностей на повідомлення, отримане від нащадка, маргіналізує таблицю по сепаратору і посилає результат. Коли нащадок отримує повідомлення від свого батька, він множить його на свою поточну таблицю умовних ймовірностей і формує таким чином свою нову таблицю. Далі він маргіналізує її (по сепаратору) і посилає своєму нащадкові. Процес триває доти, доки усе листя не отримає повідомлення. Таким чином, результатом будуть знову перераховані таблиці для кожної змінної за умови наявності спостережень, які потім необхідно нормалізувати.

Тобто знизу нагору йдуть  $\lambda$  - повідомлення, а потім зверху вниз йдуть  $\pi$  - повідомлення, у процесі проходження повідомлень відбувається перерахування значення ймовірності кліки  $\psi(Kлики_i)$ .

Далі для кожної вершини виконується пошук клік, у яких вона міститься. Якщо серед них є кліка, яка не є листом, то вибирається ця кліка. Інакше – будь-яка з них. Для знаходження ймовірності кожного стану вершини необхідно підсумувати усі значення ймовірностей для цього конкретного стану, які є в таблиці кліки.

За значенням ймовірності кліки виконується обчислення ймовірності кожної вершини кліки:

$$P(Kлики_i) = P(R_i, S_i) = P(R_i | S_i) \cdot P(S_i), \text{ то } P(R_i | S_i) = \frac{P(R_i, S_i)}{P(S_i)}. \quad (3.11)$$

Це рівняння застосовується для обчислення значень ймовірностей інстанційованих вершин [78]. Значення  $P(S_i)$  виконує роль  $\pi$  - повідомлення, а  $P(R_i | S_i)\lambda$  - повідомлення у випадку, коли  $S_i = \{ \}$ , то  $P(S_i) = 1$ . Докладнішу інформацію стосовно обчислення значень ймовірностей в об'єднаному дереві, можна знайти в роботах [65, 78, 86, 87].

### **Алгоритми визначаючого перетину**

Дж. Перл у 1988 р. [24] запропонував **алгоритм визначаючого перетину** (*cutset conditioning*), який зручно застосовувати у випадку багатозв'язаних та занадто великих мереж Байєса, коли методам поглинаючого виключення та кластеризації об'єднаного дерева не вистачає об'єму пам'яті звичайного персонального комп'ютера.

Ідея методу полягає у перетворенні багатозв'язної мережі Байєса в однозв'язну або декілька однозв'язних, застосовуючи мінімальний визначаючий перетин. Після цього застосовується алгоритм Перла для розповсюдження повідомлення по однозв'язній мережі. Кожна отримана проста, тобто однозв'язна, мережа має одну або декілька інстанційованих вершин. На рис. 3.13а наведено приклад мережі

Байеса, в якій у випадку, коли вершина  $D$  набуває певного стану, тобто стає інстанційованою, відбувається зациклення  $\alpha$  та  $\pi$  - повідомлень у підграфах  $DBSL$  і  $DLXT$ .

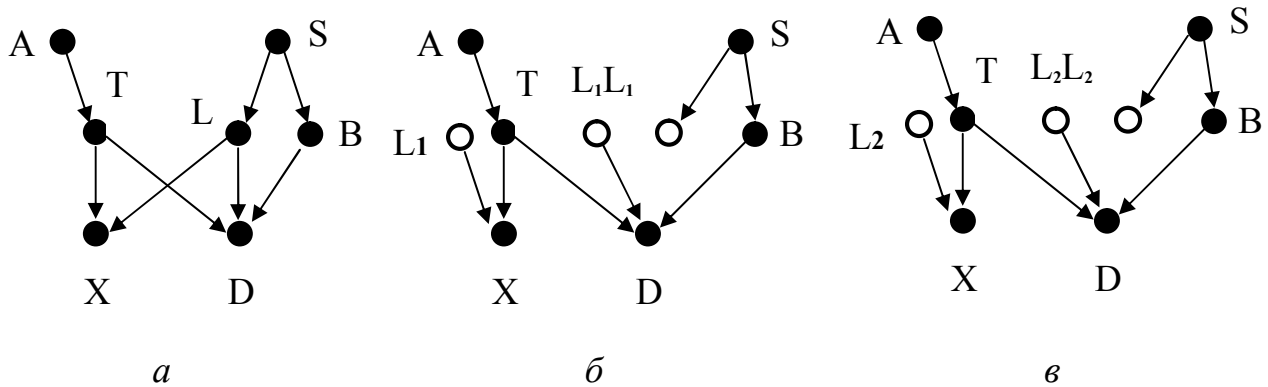


Рис. 3.13. Приклад перетворення багатозв'язної мережі (а) в однозв'язні (б, в), де визначаючий перетин складається із однієї вершини  $L$

Саме ці підграфи роблять мережу багатозв'язною, тобто потрібно позбутися циклів, які утворюють ці підграфи. Якщо вершина  $L$  може приймати один з двох станів  $\{L_1, L_2\}$ , тобто у випадку, коли вершина  $L$  стає інстанційованою, багатозв'язну мережу рис.3.13а можна записати у вигляді двох однозв'язних рис.3.13б та в.

У випадку мережі, представленої на рис. 3.13а, мінімальний визначаючий перетин складається лише з однієї вершини  $L$ . Значення довіри багатозв'язної мережі рис. 3.13а є агрегованою оцінкою, яка обчислюється за значеннями довіри в однозв'язних мережах, (рис. 3.13б та 3.13в), із врахуванням апріорної ймовірності вузла  $L$ , за формулою 3.12:

$$\begin{aligned}
 P(A, T, X, D, B, S, L) &= \sum_{L_i} (P(A, T, X, D, B, S, L = L_i) \cdot P(L = L_i)) = \\
 &= P(A, T, X, D, B, S, L = L_1) \cdot P(L = L_1) + P(A, T, X, D, B, S, L = L_2) \cdot P(L = L_2)
 \end{aligned}
 \tag{3.12}$$

Однак не все в цьому методі просто. Наприклад, у випадку, коли визначаючий перетин складається з трьох вершин і кожна вершина має вісім станів, то потрібно виконати обчислення за алгоритмом Перла розповсюдження повідомлень та обчислити достовірну оцінку апріорної ймовірності для всіх 512 структур. Тобто для цього методу саме вибір мінімального визначаючого перетину є дуже актуальною задачею [77, 88]. Але ця задача, в свою чергу, має нелінійну поліноміальну складність, для розв'язання якої використовують поглинаючі евристичні методи або ВМВ.

До методів визначаючого перетину також належать алгоритми:

1. Запропонований в 1992 році **LC-алгоритм** (*local condition*) або *алгоритм локального визначаючого перетину* [89], фактично є модифікацією алгоритму Кіма-Перла [90].
2. Опублікований в 1994 році *алгоритм глобального визначаючого перетину* (*global conditioning*) [91] є окремим випадком LS-алгоритму [56].

3. Представлений в 1995 році *алгоритм динамічного визначаючого перетину* (*dynamic conditioning*) [94] є поліпшеною версією методу визначаючого перетину [24]. Цей алгоритм має лінійну складність, у той час, як звичайний метод визначаючого перетину [24], має експоненційну складність.

4. У 2000 році запропоновано *RC-алгоритм* (*recursive conditioning*) або *алгоритм рекурсивного визначаючого перетину* [95], який увібрав у себе кращі ідеї з алгоритмів виключення змінних [69, 70] та LS [56]. Його обчислювальна складність дорівнює  $O(n \cdot \exp(w))$ , де  $n$  - кількість вершин;  $w$  - довжина ВМВ виключення змінних.

### Алгоритми виключення змінних

На початку дев'яностих років минулого століття запропоновано *VE-алгоритм* (*variable elimination algorithm*), тобто *алгоритм виключення змінних*. В основу алгоритму покладена відома ланцюгова формула декомпозиції спільного розподілу ймовірностей мережі, яка бере свій початок від теореми Байєса:

$$P(X) = P(X^{(N)} | X^{(1)}, \dots, X^{(N-1)}) \cdot P(X^{(N-1)} | X^{(1)}, \dots, X^{(N-2)}) \cdot \dots \cdot P(X^{(1)}) = \prod_{i=1}^N P(X^{(i)} | X^{(1)}, \dots, X^{(i-1)}) = \prod_{i=1}^N P(X^{(i)} | pa(X^{(i)})) \quad (3.13)$$

де  $pa(X^{(i)})$  множина батьківських вершин-батьків по відношенню до вершини  $X^{(i)}$ .

Основна ідея алгоритму виключення змінних полягає в обчисленні ймовірності вершини мережі за формулою:

$$P(X^{(n)}, e) = \sum_{X^{(k)}} \dots \sum_{X^{(3)}} \sum_{X^{(2)}} \prod_i P(X^{(i)} | pa(X^{(i)})), \quad (3.14)$$

Тобто, чим менше вершина зв'язана з  $X^{(n)}$ , тим віддаленіше вона знаходиться за межами внутрішньої суми; кожна внутрішня сума після обчислення перетворюється на нову змінну, яка надалі використовується як множник. Рис. 3.14 ілюструє приклад послідовного виключення вершин мережі Байєса

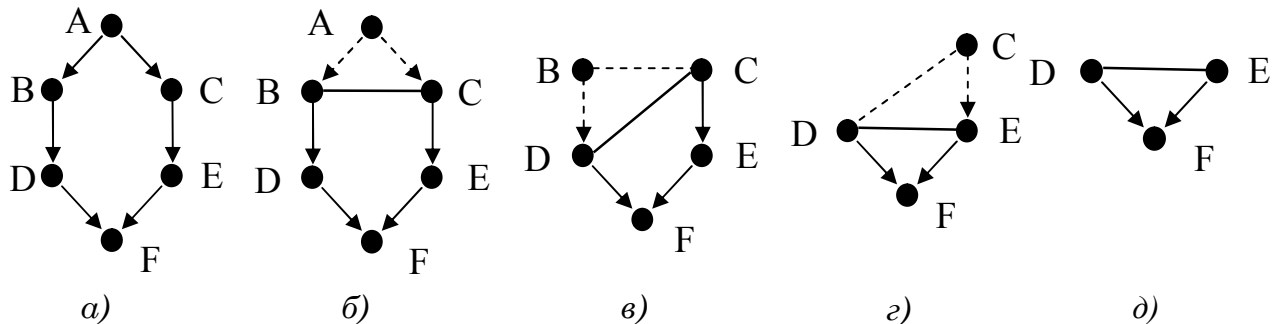


Рис. 3.14. Алгоритм виключення змінних – послідовне перетворення структури мережі Байєса: а) початкова мережа Байєса; б) виключення вершини  $A$ ; в) виключення вершини  $B$ ; г) виключення вершини  $C$ ; д) остаточний результат – кліка з трьох вершин

**Послідовність обчислень шуканої ймовірності** для відповідної мережі має наступний вид.

1. Знаходження сукупного розподілу ймовірності вершин A, B і C:  $P(A, B, C) = P(A) \cdot P(B | A) \cdot P(C | A)$ . Виключення вершини A:  $P(B, C) = \sum_A P(A, B, C)$ .
2. Знаходження сукупного розподілу ймовірностей вершин B, C, і D:  $P(B, C, D) = P(B, C) \cdot P(D | B)$ . Виключення вершини B:  $P(C, D) = \sum_B P(B, C, D)$ .
3. Знаходження сукупного розподілу ймовірностей вершин C, D і E:  $P(C, D, E) = P(C, D) \cdot P(E | C)$ . Виключення вершини C:  $P(D, E) = \sum_C P(C, D, E)$ .
4. Знаходження сукупного розподілу ймовірностей вершин D, E, і F:  $P(D, E, F) = P(D, E) \cdot P(F | D, E)$ . Виключення вершини D:  $P(E, F) = \sum_D P(D, E, F)$ .
5. Виключення вершини E:  $P(F) = \sum_E P(E, F)$ .

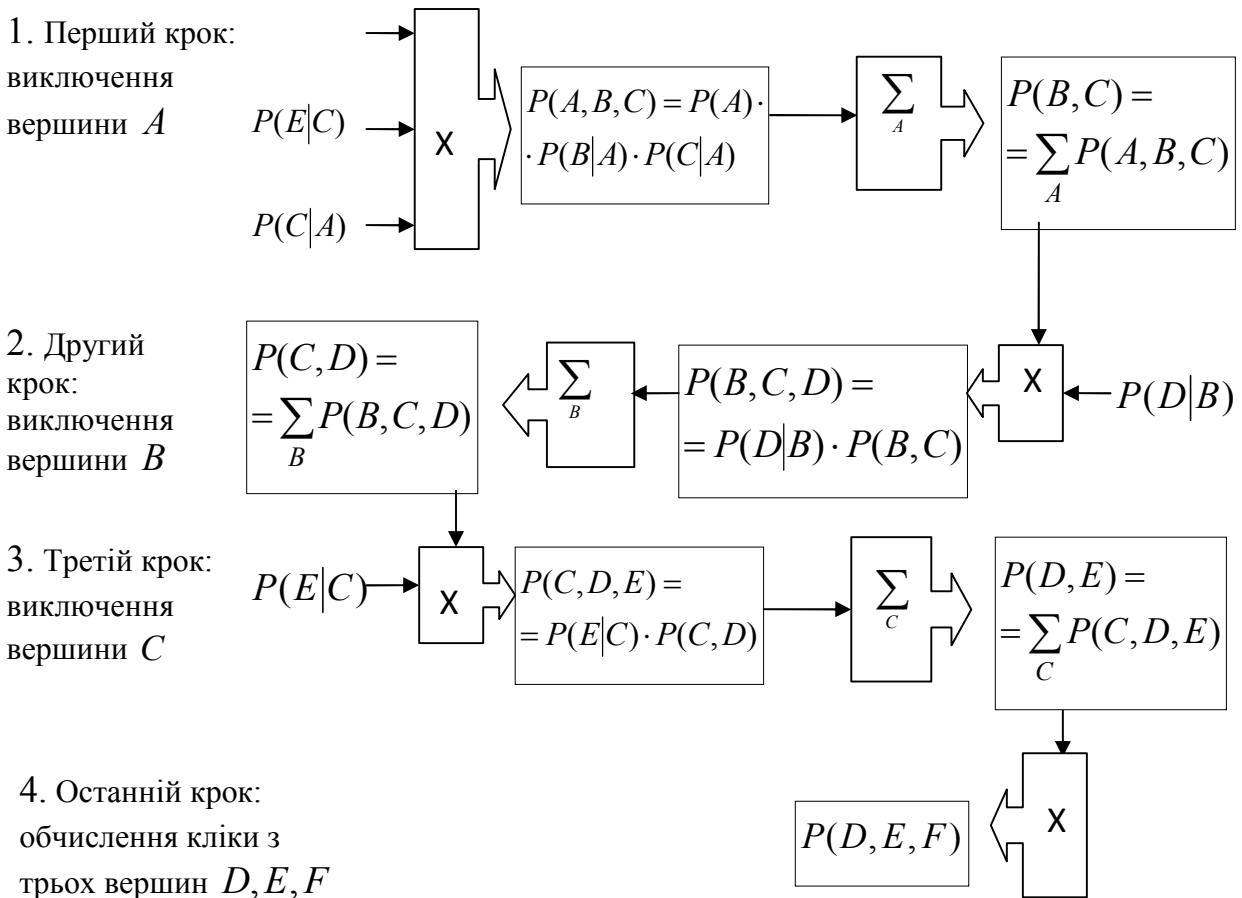


Рис. 3.15. Послідовність обчислень алгоритму виключення змінних

Докладнішу інформацію стосовно алгоритму та його практичного застосування можна знайти в літературі [69, 70, 96, 97].

Також існують різноманітні модифіковані версії цього алгоритму [98, 99], серед яких необхідно окремо відзначити **алгоритм поглинаючого виключення** (*bucket elimination*), запропонований Ріною Дехтер (Rina Dechter) [95]. Іноді в літературі він зустрічається під назвою алгоритм поглинаючого виключення для оцінювання довіри або алгоритм БЕВА (*bucket elimination for belief assessment*).

Аналогічно алгоритму виключення змінних цей метод робить послідовне виключення змінних у мережі, при цьому кожна таблиця умовних ймовірностей конвертується в  $\lambda$ -таблицю, в якій кожне значення вершини та її батька асоціюється з відповідним  $\lambda$ -значенням. На основі цього підходу розроблено цілу низку алгоритмів *elim-mp*, *elim-bel*, *elim-map*, *elim-me* та інші [95].

Як приклад, на рис. 3.16 представлена мережа Байєса, для якої відома ВМВ  $\{A, C, B, F, D, G\}$  та задано інстанційоване значення  $g=1$ , а крім того, схематично представлено приклад процесу поглинання вершин.

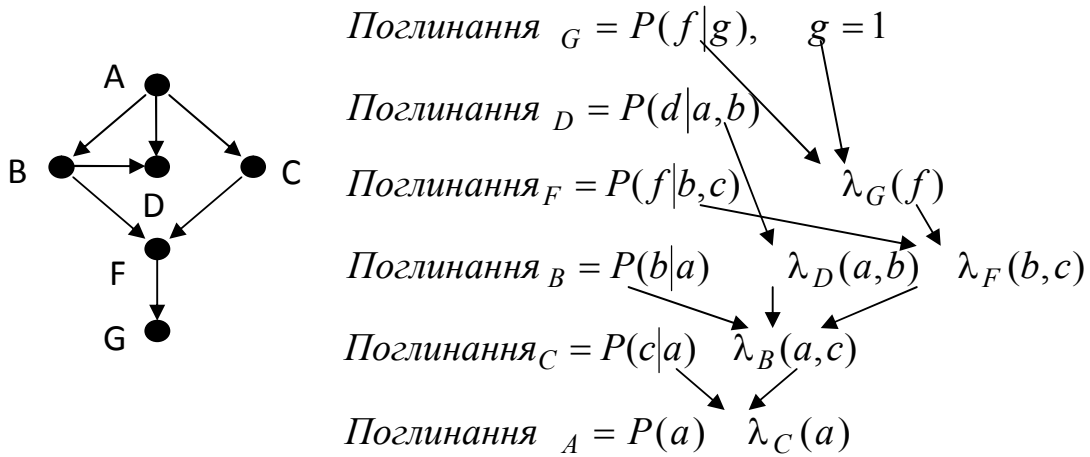


Рис. 3.16. Приклад роботи методу поглинаючого виключення

$$\begin{aligned}
 P(a, g = 1) &= P(a) \cdot \sum_f \sum_d \sum_c P(c|a) \sum_b P(b|a) \cdot P(d|a,b) \cdot P(f|b,c) \sum_{g=1} P(g|f) = \\
 &= P(a) \cdot \sum_f \lambda_G(f) \sum_d \sum_c P(c|a) \sum_b P(b|a) \cdot P(d|a,b) \cdot P(f|b,c) = \\
 &= P(a) \cdot \sum_f \lambda_G(f) \sum_d \sum_c P(c|a) \cdot \lambda_B(a,b,c,f) = P(a) \cdot \sum_f \lambda_G(f) \sum_d \lambda_C(a,d,f) = \\
 &= P(a) \cdot \sum_f \lambda_G(f) \cdot \lambda_D(a,f) = P(a) \cdot \lambda_F(a)
 \end{aligned}$$

### Алгоритми формування символічного ймовірнісного висновку

При розв'язанні практичних задач нерідко зустрічаються випадки, коли невідоме точне чисельне значення ймовірності, а відомий деякий інтервал значень. Найчастіше така ситуація зустрічається при роботі з експертами. Для вирішення даної проблеми застосовують **SPI-алгоритми** (*symbolic probabilistic inference*) або **алгоритми символічного ймовірнісного висновку**.

Свою назву ці алгоритми отримали завдяки тому, що нечіткі значення станів вершин мережі замінюють параметрами-символами, як це робиться в табл. 5.6 для вершини  $C$  [71]. Існує два підходи для здійснення символічного ймовірнісного висновку.

**Перший підхід** полягає у виконанні необхідної мінімальної кількості обчислень, тобто здійсненні тільки тих обчислень, які дійсно необхідні для побудови ймовірнісного висновку.

**Головна ідея** полягає у переході від розв'язання задачі формування ймовірнісного висновку в мережі Байєса до комбінаторної задачі оптимізації, а саме розв'язання OFP- задачі (*optimal factoring problem*), тобто **задачі оптимальної факторизації** [100, 101], яка полягає у пошуку такої факторизації  $\alpha$ , за якої  $\mu_{\alpha}(s_{\{1,2,\dots,N\}}) \rightarrow \min$ , де  $S_{\{i\}} \subset V = \{X^{(1)}, \dots, X^{(N)}\}$ , а  $\mu$  - **оціночна функція, яка визначається** так:

$$\mu(S_{\{i\}}) = 0, \text{ для } i \in [1; N], \mu(S_{I \cup J}) = \mu(S_I) + \mu(S_J) + 2^{\text{card}(S_I \cup S_J)}, \quad (3.15)$$

де  $\text{card}(S)$  - потужність множини  $S$ , тобто кількість елементів множини.

Наприклад, у випадку мережі, зображеної на рис. 3.17, для обчислення значення спільної ймовірності вершин  $d$  і  $e$  можна використовувати таку факторизацію:

$$p(d, e) = \left[ \sum_a \left[ \sum_b \left[ \sum_c [P(e|c) \cdot P(d|b,c)] \cdot P(c|a) \right] \cdot P(b|a) \right] \cdot P(a) \right] \quad (3.16)$$

У цьому випадку треба виконати 72 операції множення. Однак для цього ж випадку можна скористатись іншою факторизацією:

$$P(d, e) = \left[ \sum_c \left[ P(e|c) \left[ \sum_b P(d|b,c) \left[ \sum_a P(c|a) [P(b|a) \cdot P(a)] \right] \right] \right] \right], \quad (3.17)$$

для реалізації якої необхідно виконати лише 28 операцій множення.

При цьому вважається, що кожна вершина мережі може перебувати в одному з двох станів.

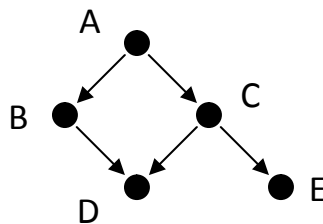


Рис. 3.17. Приклад мережі Байєса для задач OFP – оптимальної факторизації та SPI – символічного ймовірнісного висновку

Таблиця 3.6

Таблиця значень маргінальних та умовних ймовірностей вершин мережі Байєса, зображеної на рис. 3.17, із використанням параметрів-символів  $\theta$

Вершина	Параметри-символи $\theta$	
	Значення вершини = 0	Значення вершини = 1
A	$\theta_{10} = P(A = 0) = 0,2$	$\theta_{11} = P(A = 1) = 0,8$
B	$\theta_{200} = P(B = 0   A = 0) = 0,3$ $\theta_{201} = P(D = 0   A = 1) = 0,5$	$\theta_{210} = P(B = 1   A = 0) = 0,7$ $\theta_{211} = P(B = 1   A = 1) = 0,5$
C	$\theta_{300} = P(C = 0   A = 0)$ $\theta_{301} = P(C = 0   A = 1) = 0,5$	$\theta_{310} = P(C = 1   A = 0)$ $\theta_{311} = P(C = 1   A = 1) = 0,5$
D	$\theta_{4000} = P(D = 0   B = 0, C = 0) = 0,1$ $\theta_{4001} = P(D = 0   B = 0, C = 1) = 0,3$ $\theta_{4010} = P(D = 0   B = 1, C = 0) = 0,8$ $\theta_{4011} = P(D = 0   B = 1, C = 1) = 0,4$	$\theta_{4100} = P(D = 1   B = 0, C = 0) = 0,9$ $\theta_{4101} = P(D = 1   B = 0, C = 1) = 0,7$ $\theta_{4110} = P(D = 1   B = 1, C = 0) = 0,2$ $\theta_{4111} = P(D = 1   B = 1, C = 1) = 0,6$
E	$\theta_{500} = P(E = 0   C = 0) = 0,3$ $\theta_{501} = P(E = 0   C = 1) = 0,1$	$\theta_{510} = P(E = 1   C = 0) = 0,7$ $\theta_{511} = P(E = 1   C = 1) = 0,9$

*Другий підхід*, запропонований Кастілло (Castillo), полягає у використанні символічного ймовірнісного висновку, який також працює не з чисельними даними, а з параметрами-символами [71-73]. Цей метод в якості рішення видає функції, представлені у параметрично-символьній формі (табл. 3.7).

Таблиця 3.7

Таблиця значень маргінальних та умовних ймовірностей вершин мережі Байєса, зображеної на рис. 3.17 із використанням параметра-символу  $\theta_{300}$ , за умови, що вершини B та E інстанційовані ( $B=1, E=1$ )

Вершина	Значення ймовірності вершини	Значення умовної ймовірності вершини у параметрично-символьному вигляді
A	$P(A = 0) = 0,2$	$P(A = 0   B = 1, E = 1) = \frac{0,126 - 0,028 \cdot \theta_{300}}{0,446 - 0,028 \cdot \theta_{300}}$
B	$P(B = 0) = 0,46$	$P(B = 0   B = 1, E = 1) = 0$
C	$P(C = 0) = 0,42 + 0,2 \cdot \theta_{300}$	$P(C = 0   B = 1, E = 1) = \frac{0,14 - 0,098 \cdot \theta_{300}}{0,446 - 0,028 \cdot \theta_{300}}$
D	$P(D = 0) = 0,424 - 0,056 \cdot \theta_{300}$	$P(D = 0   B = 1, E = 1) = \frac{0,164 - 0,021 \cdot \theta_{300}}{0,446 - 0,028 \cdot \theta_{300}}$
E	$P(E = 0) = 0,18 + 0,04 \cdot \theta_{300}$	$P(E = 0   B = 1, E = 1) = 0$

Після цього, підставляючи замість параметрів-символів фактичні чисельні значення, виконується обчислення чисельних значень ймовірнісного висновку. Так, на рис. 3.18 продемонстрована ситуація, коли робиться заміна параметра-символу  $\theta_{300}$  з табл. 3.7 чисельним значенням 0,4.

Цей підхід також дозволяє досліджувати чутливість рішення, завдяки послідовному збільшенню або зменшенню чисельних значень параметрів-символів. Програмну реалізацію свого методу Кастілло виконував за допомогою засобів Mathematica та Maple.

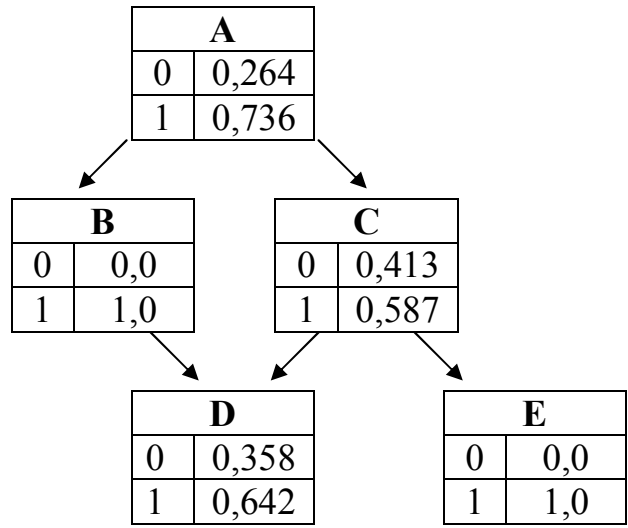


Рис. 3.18. Ймовірності станів вершин мережі Байєса, (рис. 3.17), при заміні параметра-символу в табл. 3.7 чисельним значенням  $\theta_{300} = 0,4$ .

Однак обидва підходи мають однакові проблеми: необхідність залучення спеціалізованого програмного забезпечення та велика обчислювальна складність методів, що є особливо критичним при роботі з великими мережами Байєса.

**Диференціальний підхід**

У 2000 році Дарвичем [102] було запропоновано диференціальний підхід (differential approach). Фактично, це єдиний відомий алгоритм для побудови ймовірнісного висновку із застосуванням диференціалів. Алгоритм складається з двох етапів: на першому етапі будується поліном, який описує ймовірнісний висновок; на другому етапі шукають перші та другі похідні поліному і будують ймовірнісний висновок, використовуючи їх властивості.

На першому етапі за допомогою упорядкованої множини вершин мережі Байєса представляють у вигляді поліноміального рівняння. Цей етап має розрахункову складність порядку  $O(n \cdot \exp(w))$ , де  $n$  – кількість вершин, а  $w$  – кількість станів вершин, для мережі Байєса з рис. 3.19  $n = 2, w = 2$ .

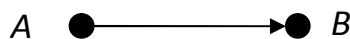


Рис. 3.19. Приклад простої мережі Байєса

Ідея використання рівнянь з параметрами-факторами за своєю суттю схожа на SPI-метод.



Таблиця 3.8

Таблиця значень ймовірності вершини  $A$  та параметрів  $\theta$  для мережі Байєса, наведеної на рис. 3.19

$A$	Значення ймовірності $P(A)$	Параметр мережі $\theta$
$\alpha$	0,3	$\theta_{\alpha} = 0,3$
$\bar{\alpha}$	0,7	$\theta_{\bar{\alpha}} = 0,7$

Таблиця 3.9

Таблиця умовних ймовірностей  $P(A|B)$  та параметрів  $\theta$  для мережі Байєса (рис. 3.19)

$A$	$B$	Значення ймовірності $P(A B)$	Параметр мережі $\theta$
$\alpha$	$b$	0,1	$\theta_{ab} = 0,1$
$\alpha$	$\bar{b}$	0,9	$\theta_{a\bar{b}} = 0,9$
$\bar{\alpha}$	$b$	0,8	$\theta_{\bar{a}b} = 0,8$
$\bar{\alpha}$	$\bar{b}$	0,2	$\theta_{\bar{a}\bar{b}} = 0,2$

Таблиця 3.10

Таблиця значень спільної ймовірності та параметризація вершин  $A$  та  $B$

$A$	$B$	$P(A,B)$	Параметризація
$\alpha$	$b$	0,03	$\lambda_{\alpha} \cdot \lambda_b \cdot 0,03$
$\alpha$	$\bar{b}$	0,27	$\lambda_{\alpha} \cdot \lambda_{\bar{b}} \cdot 0,27$
$\bar{\alpha}$	$b$	0,56	$\lambda_{\bar{\alpha}} \cdot \lambda_b \cdot 0,56$
$\bar{\alpha}$	$\bar{b}$	0,14	$\lambda_{\bar{\alpha}} \cdot \lambda_{\bar{b}} \cdot 0,14$

Таблиця 3.11

Таблиця значень індикаторів  $\lambda$  залежно від наявності інформації про стан вершин  $A$  та  $B$

Стан вершин $A$ та $B$	$\lambda_{\bar{\alpha}}$	$\lambda_{\bar{b}}$	$\lambda_b$	$\lambda_{\alpha}$
$\alpha, b$	0	0	1	1
$\alpha, \bar{b}$	0	1	0	1
$\bar{\alpha}$	0	1	1	1
Відсутня інформація про стан $A$ та $B$	1	1	1	1

Для мережі Байєса, зображеної на рис. 3.19, поліноміальне рівняння у канонічному вигляді записується як:

$$F = F(\lambda, \theta) = \sum_{i=1}^n F(\lambda_{x_i}, \theta_{f_i}) = \sum_{i=1}^n ((\prod \theta_{f_i}) \cdot (\prod \lambda_{x_i})) = \theta_{\alpha} \cdot \theta_{ab} \lambda_{\alpha} \cdot \lambda_b + \theta_{\alpha} \cdot \theta_{a\bar{b}} \cdot \lambda_{\alpha} \cdot \lambda_{\bar{b}} + \theta_{\bar{\alpha}} \cdot \theta_{\bar{a}b} \cdot \lambda_{\bar{\alpha}} \cdot \lambda_b + \theta_{\bar{\alpha}} \cdot \theta_{\bar{a}\bar{b}} \cdot \lambda_{\bar{\alpha}} \cdot \lambda_{\bar{b}} \quad (3.18)$$

де  $\lambda$  - індикатор, а  $\theta$  - параметри мережі, тобто значення ймовірності

$\theta_a = P(A = a)$ , або умовної ймовірності  $\theta_{ab} = P(B = b | A = a)$ .

При цьому за теоремою Байєса виконується рівність:

$$P(A = a, B = b) = P(B = b | A = a) \cdot P(A = a) = \theta_a \cdot \theta_{ab} \quad (3.19)$$

На другому етапі виконується обчислення перших та других похідних полінома. Оскільки для перших та других похідних полінома  $F$  виконуються властивості, які застосовують при побудові ймовірнісного висновку, то підставляючи в отримані диференціальні рівняння фактичні значення змінних та інстанційованих вершин, обчислюють відповідні значення станів вершин.

Так, для перших похідних  $\frac{\partial F}{\partial \lambda_{x_i}}$  та  $\frac{\partial F}{\partial \theta_{f_i}}$  виконуються такі властивості:

$$\frac{\partial F(\lambda, \theta)}{\partial \lambda_x} = P(x, e - X | \theta) \quad \text{та} \quad \frac{\partial F(e, \theta)}{\partial \theta_f} \theta(f) = P(f, e | \theta), \quad (3.20)$$

де  $f$  - це множина вершин мережі (family);  $e$  - множина інстанційованих станів;  $X$  - вершина мережі Байєса. Вираз  $e - X$  означає підмножину  $e$ , в якій відсутня  $X$ ; наприклад, якщо  $e = \{a, b, \bar{c}\}$ , то виконуються умови:  $e - A = \{b, \bar{c}\}$  та  $e - AC = \{b\}$ .

Тобто чисельне значення  $\frac{\partial F}{\partial \lambda_{x_i}}$  - це ймовірність для вершини  $X$  в  $i$ -му стані.

Також для кожної вершини  $X$  та множини інстанційованих станів  $e$  виконуються рівняння:

$$P(e - X) = \sum_x \frac{\partial F(e)}{\partial \lambda_x} \quad \text{та} \quad P(x | e - X) = \frac{\partial F(e) / \partial \lambda_x}{\sum_x \partial F(e) / \partial \lambda_x}. \quad (3.21)$$

Наприклад, для мережі Байєса (рис. 3.19) у випадку, коли  $e = a$  маємо:

$$P(e - A) = \frac{\partial F(e)}{\partial \lambda_a} + \frac{\partial F(e)}{\partial \lambda_{\bar{a}}} = 1, \quad P(\bar{a} | e - A) = \frac{\partial F(e) / \partial \lambda_{\bar{a}}}{\partial F(e) / \partial \lambda_a + \partial F(e) / \partial \lambda_{\bar{a}}} = \frac{0,7}{1} = 0,7.$$

Для других похідних, за умови що  $X \neq Y$  та  $F_i \neq F_j$ , виконуються властивості:

$$\frac{\partial^2 F(e, \theta)}{\partial \lambda_x \cdot \partial \lambda_y} = P(x, y, e - XY | \theta); \quad (3.22)$$

$$\frac{\partial^2 F(e, \theta)}{\partial \lambda_x \cdot \partial \theta_{f_i}} \cdot \theta(f_i) = P(x, f_i, e - X | \theta); \quad (3.23)$$

$$\frac{\partial^2 F(e, \theta)}{\partial \theta_{f_i} \cdot \partial \theta_{f_j}} \cdot \theta(f_i) \cdot \theta(f_j) = P(f_i, f_j, e | \theta). \quad (3.24)$$

**Обчислювальна складність** при знаходженні частинних похідних  $\partial F / \partial \lambda_{x_i}$

та  $\partial F / \partial \theta_{f_i}$  дорівнює  $O(n \cdot \exp(w))$ , а при  $\frac{\partial^2 F}{\partial \lambda_{x_i} \cdot \partial \theta_{f_i}}$  дорівнює  $O(n^2 \cdot \exp(w))$ .

### Інші алгоритми формування точного висновку

У 1989 році Р. Шехтер [75] запропонував **AR-алгоритм** від англійського *arc reversal*, тобто *реверсування дуг*. Іноді AR-алгоритм можна зустріти під назвою *зменшення кількості вершин (node reduction)*. Цей алгоритм є одним із найперших алгоритмів точного висновку. Він виконує модифікацію мережі Байєса шляхом послідовного реверсування дуг, застосовуючи теорему Байєса та інформацію про інстанційованні вершини. Л. Гааг у 1993 році описав **алгоритм поглинання подій (evidence absorption)** [103], який є симбіозом *алгоритму Перла* [59] та *AR-алгоритму* [75]. Тобто, до категорії “інших” належать алгоритми, що є комбінацією декількох алгоритмів різних типів.

### Апроксимаційні алгоритми формування висновку

На практиці, при аналізі задач, які складаються з сотень, тисяч а іноді і десятків тисяч факторів, з’являється потреба формування ймовірнісного висновку у великих мережах Байєса, які складаються з тисяч вершин. Для таких мереж алгоритми формування точного висновку не працюють у зв’язку з великою обчислювальною складністю, яка близька до експоненційної. Саме тому були розроблені апроксимаційні алгоритми, які, жертвуючи точністю обчислень, роблять можливим вирішення задач великої розмірності. Апроксимаційні алгоритми поділяються на:

- алгоритми стохастичної вибірки;
- алгоритми неповного висновку;
- варіаційні алгоритми;
- пошукові алгоритми.

Одним із прикладів практичного застосування апроксимаційних алгоритмів можна навести систему медичної діагностики (*QMR – швидка медична довідка (Quick Medical Reference)*), яка представлена на рис.3.20, її інформаційна база складається із статистичних та експертних даних [104]. QMR-система почала розроблятися у Пітсбургському університеті у 1980 році, а пізніше увійшла до складу системи Internist-I [104] як один із діагностичних інструментів лікаря-терапевта. Розробку системи Internist-I було розпочато на початку 1970-х років, а першу версію закінчено в 1974 р. Сьогодні мережа Байєса QMR-системи складається приблизно з 6000 вершин, з’єднаних більш ніж 415000 дугами. Система спроможна розпізнати близько 750 видів різних захворювань за більше ніж 5000 симптомами та результатами лабораторних аналізів.

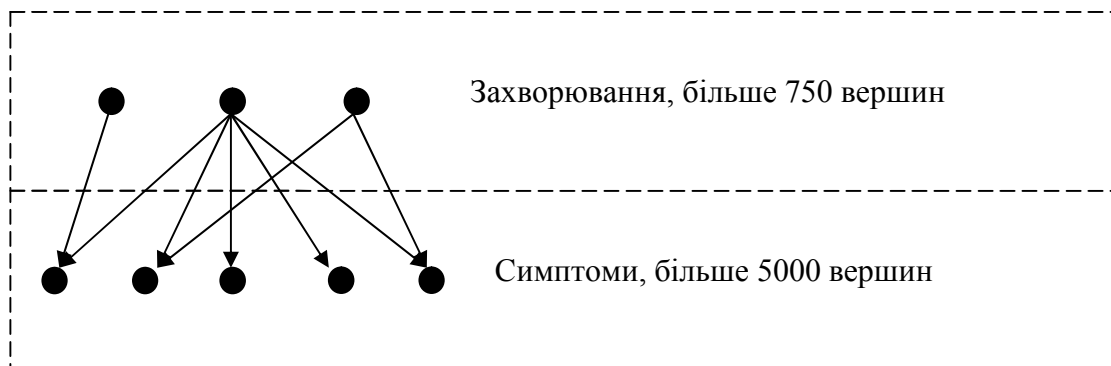


Рис. 3.20. Мережа Байєса для QMR системи (швидка медична довідка)

Аналізуючи QMR-систему, можна зрозуміти необхідність переходу від точних методів побудови ймовірнісного висновку до апроксимаційних, жертвуючи при цьому точністю обчислень. Однак цей крок дозволяє розширити межі застосування байєсівських мереж.

### Алгоритми стохастичного вибору

Серед апроксимаційних алгоритмів саме алгоритми стохастичного вибору найчастіше використовують для формування ймовірнісного висновку. У літературі ця група методів зустрічається під назвами *стохастичної дискретизації (stochastic sampling)* або *стохастичного моделювання (stochastic modeling)*.

Алгоритми стохастичної дискретизації поділяють на дві групи:

- дискретизації вибірки за значущістю;
- методи Монте-Карло для ланцюгів Маркова (МКМЛ).

Ідея *дискретизації вибірки за значущістю (importance sampling)* ґрунтується на припущенні, що деякі значення випадкової величини в процесі моделювання мають більшу значущість (ймовірність) для оцінюваної функції (параметра), ніж інші. Якщо ці "ймовірніші" значення будуть з'являтися в процесі вибору випадкової величини частіше, дисперсія оцінюваної функції зменшиться. Отже, базова методологія дискретизації вибірки за значущістю полягає у виборі розподілу, що сприяє вибору "ймовірніших" значень випадкової величини. Такий "зміщений" розподіл змінює оцінювану функцію, якщо він застосовується безпосередньо в процесі розрахунку. Однак результат розрахунку змінюється відповідно до цього зміщеного розподілу і це гарантує, що нова оціночна функція вибірки за значущістю не буде зміщеною.

*Методи Монте-Карло для ланцюгів Маркова (МКЛМ)* – поширений клас методів для побудови ймовірнісного висновку в байєсівській мережі; в літературі часто зустрічається під назвою *МСМС (Markov chain Monte Carlo)* [105]. Методи Монте-Карло – це загальна назва групи чисельних методів, які ґрунтуються на генеруванні великої кількості реалізацій випадкового процесу, що формується таким чином, щоб його ймовірнісні характеристики збігалися з аналогічними величинами розв'язуваного завдання. Однією з реалізацій *методу Монте-Карло* є *алгоритм Метрополіса (Metropolis algorithm)*, який запропоновано у 1953 р. в роботі [106].

В свою чергу, у 1984 р. на основі *алгоритму Метрополіса-Хастінга (Metropolis-Hastings algorithm)*, який є модифікацією *алгоритму Метрополіса*, запропоновано метод *Монте-Карло для ланцюгів Маркова* для мереж Байєса [107].

Наприклад, якщо мережа Байєса складається з множини вершин  $X = \{X_1, \dots, X_n\}$ , то математичне сподівання функції  $a(X_1, \dots, X_n)$ , яка описує розподіл множини  $X$ , записується апроксимаційним рівнянням:

$$\begin{aligned} \langle a \rangle &= E(a(X)) = \sum_{\tilde{x}_1} \dots \sum_{\tilde{x}_n} a(\tilde{x}_1, \dots, \tilde{x}_n) \cdot P(X_1 = \tilde{x}_1, \dots, X_n = \tilde{x}_n) \approx \\ &\approx \frac{1}{N} \cdot \sum_{t=0}^{N-1} a(x_1^{(t)}, \dots, x_n^{(t)}) \end{aligned} \quad , \quad (3.25)$$

де  $N$  - розмір вибірки;  $x_1^{(t)}, \dots, x_n^{(t)}$  – значення вершин на  $t$ -му кроці.

Серед різних варіацій алгоритмів Монте-Карло для ланцюгів Маркова найпопулярнішими є алгоритми дискретизації Гіббса (Gibbs sampler)[108] та Метрополіса-Хастінгса (Metropolis-Hastings) [109].

Окрім алгоритмів вибірки за значущістю та методів Монте-Карло для ланцюгів Маркова вирізняють наступні **методи стохастичної дискретизації**:

- **алгоритм логічного відбору** (*logic sampling*), запропонований у 1986 р. Генріоном (Henrion) [111], є одним із найпростіших алгоритмів прямої вибірки. Ідея алгоритму полягає у багаторазовому моделюванні вибірки за мережею проходу за напрямком зв'язків між вершинами, відкиданні елементів вибірки, що протирічать спостереженням і оцінці ймовірностей шуканих змінних за допомогою частоти появи відповідних подій у вибірці. Якщо вирішується задача без спостережень, то алгоритм логічної вибірки працює дуже добре, але у випадку малої ймовірності спостереження з'являються проблеми. Частка правильних елементів вибірки експоненційно зменшується залежно від кількості спостережень.

- **алгоритм правдоподібного зважування** (*LW – likelihood weighting*) – запропонований в 1989 р. Р. Фангом та К. Чангом [112], удосконалює алгоритм логічної вибірки [113]. Він працює швидше навіть у більших мережах, але проблема малої ймовірності подій все одно залишається.

- **алгоритм зворотної дискретизації** (*backward sampling*) запропонований Р. Фангом та Б. Фаверо в 1994 р. [114]. Ідея цього методу, як і алгоритму правдоподібного зважування [110, 112, 114], полягає в обчисленні значень ймовірностей станів вершин із застосуванням частоти появи відповідних подій, які відповідають конкретному стану вершини у навчальній множині моделей.

- **адаптивний алгоритм дискретизації з відхиленням** (*ARS – adaptive rejection sampling*), запропонований у 1992 р. В. Гілксом та П. Вальдом [115, 116] і удосконалений у 2002 р. К. Кохненом [117], застосовується для мереж Байєса, в яких функції розподілу вершин описуються логарифмічними функціями.

- **алгоритм BN-RAS** (*Bayesian network randomized approximation scheme*) – запропонований у 1990 р. Р. Чавесом та Дж. Купером [195] демонструє прийнятні результати при роботі з великими мережами Байєса, в яких наявна велика кількість дуг між вершинами.

- **алгоритми SIS** (*self-importance sampling*) та **HIS** (*heuristic importance sampling*), розроблені Р. Шахтером та М. Питом ще в 1990 р. [118], відрізняються між собою методом генерування вибірки за значущістю: SIS використовують оціночну функцію, а HIS – модифікований алгоритм розповсюдження повідомлення за однозв'язаним деревом.

- **алгоритм AIS** (*adaptive importance sampling*) розробили Дж. Ченг та М. Друздзел у 2000 році [119]. У цьому алгоритмі при генеруванні навчальних даних на різних етапах навчання застосовують різні вагові коефіцієнти. У 2001 році, завдяки введенню в AIS нового правила зупинки, запропоновано ефективніший алгоритм **AIS-BN** [120].

Всі алгоритми стохастичної дискретизації відрізняються один від одного функціями генерування вибірки, принципом побудови вагових коефіцієнтів цих функцій та засобами генерування множини випадкових моделей для відповідного типу розподілу.

### **Алгоритми формування часткового або неповного висновку**

Дану групу методів іноді називають по-іншому, *методами спрощення структури (model simplification methods)*. Ідея цих методів полягає у послідовній зміні структури мережі Байєса таким чином, щоб зменшити її структурну складність настільки, щоб стало можливим застосування точних методів формування висновку. Наприклад, Ф. Дженсен та С. Андерсен [121] пропонували анігілювати маленькі (незначні) ймовірності, а Кјаерулф U. [122] та R. Engelen [123] – позбавлятися дуг, які відображають слабкий зв'язок між вершинами. Зрозуміло, що отримане таким чином значення ймовірнісного висновку, буде відрізнятися від реального. Існує багато методів формування часткового висновку, зокрема:

- **алгоритм визначаючого обмеження (bounded conditioning)** [124] – один із перших алгоритмів, запропонований у 1988 р. Horvitz E.J., Suermondt H.J. and Cooper G.F. [125], доповнений у 1994 р. Draper D. and Hanks S. [126] За своєю ідеєю він схожий на метод визначаючого перетину [24] в тому, що для зменшення обчислювальної складності необхідно позбавляються циклів всередині мережі Байєса, визначивши множину вершин, які утворюють циклічні перетини (cycle cutset);

- **алгоритм локального часткового оцінювання** або **LPE (localized partial evaluation)** [126]. При обчисленні ймовірностей станів вершини алгоритм ігнорує частину мережі, а саме ті вершини, які знаходяться достатньо далеко від неї;

- **покроковий алгоритм SPI (incremental SPI)**, представлений Д'Амбросіо в 1993 р., є модифікованою версією алгоритму точного символічного висновку для роботи з великими мережами Байєса [127];

- ідея **алгоритму часткового оцінювання ймовірності (probabilistic partial evaluation)** [128] полягає в тому, щоб поліпшити процедуру обчислення ймовірностей моделі завдяки ігноруванню відмінностей між схожими ймовірностями; особливістю алгоритму є використання поняття батьківського контексту (parent context), яке дозволяє робити декомпозицію ймовірнісного розподілу по аналогії з SPI-алгоритмом та застосування VE або BEVA алгоритмів, які за упорядкованою множиною вершин роблять згортку параметрів, що, у свою чергу, дозволяє визначати та ігнорувати відмінності між ймовірностями;

- **алгоритм mini-bucket** запропонувала Р. Дехтер в 1997 р. [129] як модифікацію алгоритму поглинаючого виключення [95], згодом на основі даного алгоритму були розроблені алгоритми *approx-opt* та *approx-bel-max*;

- в алгоритмі **Sarkar** [130] для зменшення обчислювальної складності запропоновано апроксимацію структури мережі, а саме перехід від байєсівської мережі до байєсівського дерева;

- **алгоритм зменшення розміру таблиці умовних ймовірностей у просторі станів (state space abstraction algorithm)** [131] передбачає, що для зменшення обсягів обчислень і спрощення моделі слід зменшити розмір таблиці умовних ймовірностей кожної вершини.

### **Варіаційні алгоритми**

Головна ідея варіаційних алгоритмів полягає в усередненні значень ймовірностей вершин, тобто при обчисленні розглядаються тільки значущі величини, а всі інші не враховуються. Для кожної вершини мережі розривають

зв'язки з іншими вершинами і вводять нові варіаційні параметри. Завдяки послідовній зміні значень варіаційних параметрів досягається мінімізація перехресної ентропії між апроксимаційним та фактичним значенням ймовірності. Мережа трансформується в підграф початкового графу, в якому деякі вершини позбавляються зв'язків, доки не буде можливо застосувати точний алгоритм ймовірнісного висновку.

**Алгоритм VB** (*variational Bayesian*) описано Ghahramani Z. G. and Beal M. J. [132], а докладний опис роботи різних варіаційних алгоритмів при формуванні ймовірнісного висновку для QMR-системи – у роботах Jordan M., Jaakkola T.S. та ін. [104, 133].

### Пошукові алгоритми

Основою пошукових алгоритмів (search-based) є ідея переходу від задачі ймовірнісного висновку до оптимізаційної задачі пошуку найбільш ймовірного значення. Вважається, що найбільш ймовірну частку мережі складає порівняно невелика кількість вершин, і, виконавши пошук найбільш ймовірних інстанціювань, можна отримати границі апостеріорних ймовірностей значень вершин. Тобто пошукові алгоритми ґрунтуються на евристичних алгоритмах пошуку, які використовують при переході від розв'язання задачі ймовірнісного висновку до оптимізаційної задачі. За своєю метою ця група алгоритмів схожа на алгоритми стохастичної дискретизації: у даному випадку також вирішується задача пошуку найбільш ймовірного висновку [134]. Дана група алгоритмів широко використовується в діагностиці – медичній та інженерно-технічній.

На базі цього підходу Дж. Купер у 1984 р. розробив медичну інформаційну систему NESTOR [135], в 1991 р. Henrion запропонував *Top-N* метод [111], а в 1993 р. Poole розробив Conflict – метод медичної діагностики [136]. У 1994 році Драздел теоретично обґрунтував, що асиметрія сукупного розподілу ймовірностей мережі залежить від асиметрій таблиць умовних ймовірностей. Однак емпіричні результати Lin та Druzdel показали, що хоча невелика частина всієї кількості інстанціювань справді займає найбільш ймовірну частку мережі, вона все одно дуже велика.

У 1995 році створено алгоритм з використанням генетичного підходу для паралельних обчислень у системі OVERMIND, яка, в свою чергу, є частиною он-лайн експертної системи PESKI, призначеної для інженерно-технічної діагностики в космічній програмі Space Shuttle [137].

У 1998 році запропоновано алгоритм для мереж Байєса з великою кількістю дуг [138], який ґрунтується на ІВ-підході (independence-based), ідея якого полягає в агрегуванні ймовірностей з метою зменшення об'єму обчислень.

### 3.3. Формування ймовірнісного висновку у байєсівській мережі на основі навчальних даних

Для формування ймовірнісного висновку вхідними даними є такі:

1. Множина навчальних даних  $D = \{d_1, \dots, d_n\}$ , де  $d_i = \{x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(N)}\}$  (нижній індекс – номер спостереження, а верхній – номер змінної),  $n$  – кількість спостережень. Кожне спостереження складається з  $N (N \geq 2)$  змінних

$X^{(1)}, X^{(2)}, \dots, X^{(N)}$ , кожна  $j$ -а змінна ( $j=1, \dots, N$ ) має  $A^{(j)} = \{0, 1, \dots, \alpha^{(j)} - 1\}$ , де  $(\alpha^{(j)} \geq 2)$  станів.

2. Структура мережі Байєса  $G$  представлена множиною з  $N$  предків  $(\Pi^{(1)}, \dots, \Pi^{(N)})$ , тобто для кожної вершини  $j=1, \dots, N$ ;  $\Pi^{(j)}$  – множина батьківських вершин, при цьому  $\Pi^{(j)} \subseteq \{X^{(1)}, \dots, X^{(N)}\} \setminus X^{(j)}$  (вершина не може бути батьком для самої себе, тобто петлі в графі відсутні).

3. Множина інстанційованих вершин  $\{X^{(P_1)} = x^{(P_1)}, \dots, X^{(P_v)} = x^{(P_v)}\}$ , тобто вершин, що перебувають у деякому певному стані з одиничною ймовірністю. Якщо множина інстанційованих вершин порожня, то потрібно використати ймовірнісний висновок, який ґрунтується на класичній теоремі Байєса.

Формування висновку за методом ймовірнісного висновку на основі навчальних даних відбувається покроково:

*Крок 1.* За множиною навчальних даних обчислюється матриця емпіричних значень спільного розподілу ймовірностей всієї мережі  $P(X^{(1)}, \dots, X^{(N)})$  за формулою:

$$P_{matrix}(X^{(1)} = x^{(1)}, \dots, X^{(N)} = x^{(N)}) = \frac{n[X^{(1)} = x^{(1)}, \dots, X^{(N)} = x^{(N)}]}{n}, \quad (3.26)$$

де  $n$  – кількість навчальних спостережень,  $x^{(j)} \in A^{(j)}$ , а

$$n[X^{(1)} = x^{(1)}, \dots, X^{(N)} = x^{(N)}] = \sum_{j=1}^n I(X^{(1)} = x_j^{(1)}, \dots, X^{(N)} = x_j^{(N)}), \quad (3.27)$$

де функція  $I(E)=1$ , якщо предикат  $E=true$ , інакше  $I(E)=0$ .

Алгоритм обчислення значень ймовірностей всіх можливих станів неінстанційованих вершин:

```

for  $j=1$  to  $N$  if  $X^{(j)} \notin \{X^{(P_1)}, \dots, X^{(P_v)}\}$  then
  begin
     $sum=0$ ;
     $\forall x^{(j)} \in A^{(j)}$  do
      begin
        for  $k=1$  to  $last\_string\_matrix$  do
          begin
            if  $(X_{matrix}^{(P_1)} = x^{(P_1)})$  and ... and  $(X_{matrix}^{(P_v)} = x^{(P_v)})$ 
              and  $(X_{matrix}^{(j)} = x^{(j)})$  then
                begin
                   $P(X^{(j)} = x^{(j)}) = P(X^{(j)} = x^{(j)}) + P_{matrix}(X_{matrix}^{(1)}, \dots, X_{matrix}^{(N)});$ 
                end;
              end;
            end;
           $sum=sum + P(X^{(j)} = x^{(j)});$ 
        end;
      end;
    end;
  end;

```



```

 $\forall x^{(j)} \in A^{(j)} \text{ do}$ 
  begin
     $P(X^{(j)} = x^{(j)}) = \frac{P(X^{(j)} = x^{(j)})}{sum};$ 
  end;
end;

```

Далі обчислюються значення ймовірностей всіх можливих станів, неінстанційованих вершин за даним за алгоритмом.

*Крок 2.* Перебираємо послідовно всі вершини мережі Байєса. Якщо вершина не інстанційована, то потрібно обчислити значення ймовірностей всіх можливих станів цієї вершини. Для цього виконується послідовний перебір усіх рядків матриці емпіричних значень спільного розподілу ймовірностей всієї мережі. Якщо значення вершин рядка збігаються зі значеннями інстанційованих вершин і станом аналізованої вершини, то відповідне значення  $P_{matrix}(X^{(1)}, \dots, X^{(N)})$  додається до значення ймовірності відповідного стану аналізованої вершини. Після цього нормуються значення ймовірностей станів аналізованої вершини. *Вихідними даними* є значення ймовірностей всіх можливих станів всіх неінстанційованих вершин.

**Приклад роботи методу ймовірнісного висновку в мережі Байєса на основі навчальних даних.**

Розглянемо побудову ймовірнісного висновку з використанням класичної теореми Байєса та на основі навчальних даних.

**Вхідні дані задачі:**

1. Множина навчальних даних з табл. 3.1.

Таблиця 3.12

Набір з 10 навчальних спостережень для мережі Байєса

Номер спостереження	Відповіді на навчальні дані			
	S	C	B	D
1	1	1	1	1
2	1	0	1	0
3	1	0	0	0
4	0	0	1	1
5	0	0	0	0
6	1	0	0	0
7	1	1	1	1
8	0	0	0	0
9	0	0	0	1
10	0	0	0	0

2. Структура мережі Байєса, наведена на рис. 3.1.

Структура мережі, що відповідає даним з таблиці 3.1 (рис. 3.21).

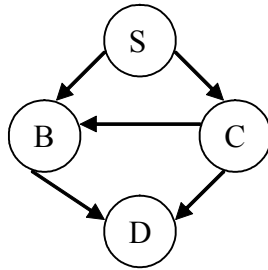


Рис. 3.21 Структура мережі Байєса, що відповідає даним табл. 3.1

3. Множина інстанційованих вершин складається з двох вершин:  $\{S = 0; C = 0\}$ , тобто пацієнт не палить і не хворий на рак:  $P(S = 0) = 1$  та  $P(C = 0) = 1$ .

**Визначити:** ймовірність наявності у пацієнта задишки та бронхіту:  $P(B) = ?$ ;  $P(D) = ?$

*Побудова ймовірнісного висновку з використанням класичної теореми Байєса.*

*Вхідні дані:* множина навчальних даних (дані з таблиці 3.1.) При обчисленні використовуються таблиці умовних ймовірностей (рис. 3.22):

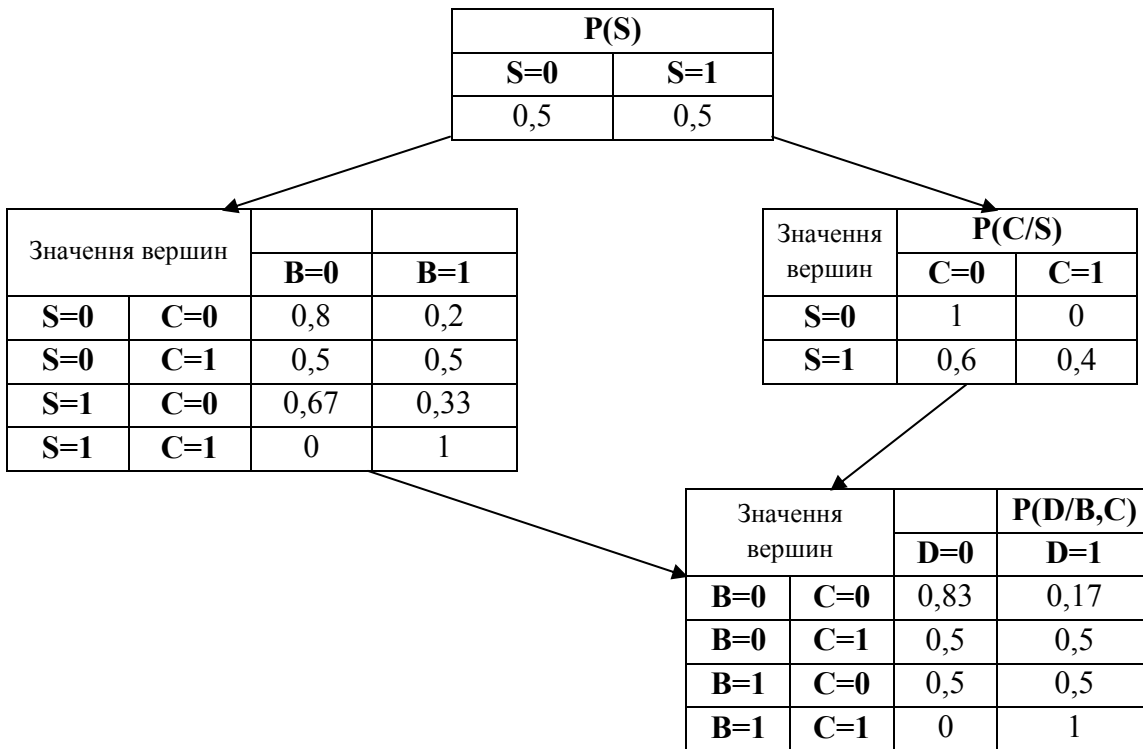


Рис. 3.22. Мережа Байєса у вигляді таблиць умовних ймовірностей вершин, що відповідають даним табл. 3.1

*Обчислення.* Для розрахунку ймовірності вершин "задишка" й "бронхіт" застосуємо звичайний алгоритм прямого розповсюдження інформації по мережі Байєса із використанням теореми Байєса.

$$P(B = 0) = P(B = 0 | S = 0, C = 0) \cdot P(S = 0, C = 0) = 0,8 \cdot 1 = 0,8;$$

$$P(B = 1) = P(B = 1 | S = 0, C = 0) \cdot P(S = 0, C = 0) = 0,2 \cdot 1 = 0,2;$$

$$\begin{aligned} P(D = 0) &= P(D = 0, B, C = 0) = P(D = 0, B = 0, C = 0) + P(D = 0, B = 1, C = 0) = \\ &= P(D = 0 | B = 0, C = 0) \cdot P(B = 0 | C = 0) \cdot P(C = 0) + P(D = 0 | B = 1, C = 0) \cdot P(C = 0) = \\ &= 0,83 \cdot 0,8 \cdot 1 + 0,5 \cdot 0,2 \cdot 1 = 0,764; \end{aligned}$$

$$\begin{aligned} P(D = 1) &= P(D = 1, B, C = 0) = P(D = 1, B = 0, C = 0) + P(D = 1, B = 1, C = 0) = \\ &= P(D = 1 | B = 0, C = 0) \cdot P(B = 0 | C = 0) \cdot P(C = 0) + P(D = 1 | B = 1, C = 0) \cdot P(C = 0) \cdot \\ &\cdot P(B = 1 | C = 0) \cdot P(C = 0) = 0,17 \cdot 0,8 \cdot 1 + 0,5 \cdot 0,2 \cdot 1 = 0,236 \end{aligned}$$

*Результат.* За умови, що пацієнт не палить і не хворий на рак, ймовірність наявності у нього бронхіту дорівнює 20%, а задишки 23,6%.

*Побудова ймовірнісного висновку на основі навчальних даних.*

*Обчислення ймовірності вершин "задишка" й "бронхіт":*

$$P(B = 0) = P(S = 0, C = 0, B = 0, D = 0) + P(S = 0, C = 0, B = 0, D = 1) = 0,4;$$

$$P(B = 1) = P(S = 0, C = 0, B = 1, D = 0) + P(S = 0, C = 0, B = 1, D = 1) = 0,1;$$

$$sum = P(B = 0) + P(B = 1) = 0,5;$$

$$P(B = 0) = \frac{P(B = 0)}{sum} = 0,8; \quad P(B = 1) = \frac{P(B = 1)}{sum} = 0,2;$$

$$P(D = 0) = P(S = 0, C = 0, B = 0, D = 0) + P(S = 0, C = 0, B = 1, D = 0) = 0,3;$$

$$P(D = 1) = P(S = 0, C = 0, B = 0, D = 1) + P(S = 0, C = 0, B = 1, D = 1) = 0,2;$$

$$sum = P(D = 0) + P(D = 1) = 0,5;$$

$$P(D = 0) = \frac{P(D = 0)}{sum} = 0,6; \quad P(D = 1) = \frac{P(D = 1)}{sum} = 0,4.$$

*Результат.* З отриманих обчислень випливає, що якщо пацієнт не палить і не хворий на рак, ймовірність наявності в нього бронхіту дорівнює 20%, а задишки 40%.

*Аналіз отриманих результатів.* Ймовірнісний висновок, який ґрунтується на прямому розповсюдженні інформації із використанням теореми Байєса щодо наявності у пацієнта задишки, дав чисельний результат 23,6%, а метод ймовірнісного висновку на основі навчальних даних - 40%. Значення ймовірності для вершини, що описує задишку для різних методів відрізняються в силу того, що при формуванні таблиць умовних ймовірностей на експериментальних даних відбувається часткова втрата інформації. Але необхідно зазначити, що запропонований метод формування ймовірнісного висновку на основі навчальних даних дає точний чисельний результат, який точно співпадає з навчальними даними з табл. 3.1.

### 3.4. Методи оцінювання якості побудови ймовірнісного висновку

Для оцінювання якості роботи будь-якого методу побудови ймовірнісного висновку можна скористатися такими величинами: середньоквадратична похибка, *KL*-відстань або квадратична відстань Хеллінджера (Hellinger) [88].

*Середньоквадратична похибка MSE :*

$$MSE = \frac{1}{\sum_{X^{(i)} \in X \setminus E} \text{card}(A^{(i)})} \cdot \sum_{X^{(i)} \in X \setminus E} \left( \sum_{\forall x^{(i)} \in X^{(i)}} (P(x^{(i)} | e) - \hat{P}(x^{(i)} | e))^2 \right), \quad (3.28)$$

*KL-відстань  $D_K$  між значенням ймовірності  $P(X^{(i)} | e)$  та оцінкою  $\hat{P}(X^{(i)} | e)$ :*

$$D_K(P(X^{(i)} | e), \hat{P}(X^{(i)} | e)) = \sum_{\forall x^{(i)} \in A^{(i)}} \left( P(x^{(i)} | e) \cdot \log \left( \frac{P(x^{(i)} | e)}{\hat{P}(x^{(i)} | e)} \right) \right). \quad (3.29)$$

*KL-відстань  $D_K$  всієї мережі Байєса:*

$$D_K(P, \hat{P}) = \frac{1}{\text{card}(X \setminus E)} \cdot \sum_{X^{(i)} \in X \setminus E} D_K(P(X^{(i)} | e), \hat{P}(X^{(i)} | e)) \quad (3.30)$$

*Квадратична відстань Хеллінджера  $D_H$  між значенням ймовірності  $P(X^{(i)} | e)$  та оцінкою  $\hat{P}(X^{(i)} | e)$ :*

$$D_H(P(X^{(i)} | e), \hat{P}(X^{(i)} | e)) = \sum_{\forall x^{(i)} \in A^{(i)}} \left( \sqrt{P(x^{(i)} | e)} - \sqrt{\hat{P}(x^{(i)} | e)} \right)^2. \quad (3.31)$$

*Квадратична відстань Хеллінджера  $D_H$  всієї мережі Байєса:*

$$D_H(P, \hat{P}) = \frac{1}{\text{card}(X \setminus E)} \cdot \sum_{X^{(i)} \in X \setminus E} D_H(P(X^{(i)} | e), \hat{P}(X^{(i)} | e)). \quad (3.32)$$

У наведених формулах  $X = \{X^{(1)}, \dots, X^{(N)}\}$  - множина всіх вершин графа мережі Байєса  $G$ , де кожна  $j$ -а вершина мережі ( $j = 1, \dots, N$ ) має  $A^{(j)} = \{0, 1, \dots, \alpha^{(j)} - 1\}$ , ( $\alpha^{(j)} \geq 2$ ) станів; запис  $\text{card}(A^{(j)})$  означає потужність множини  $A^{(j)}$  (кількість елементів, з яких складається множина);  $E \subset X$ ,  $E = e$  множина подій (інстанційовані вершини);  $P(X^{(i)} | e)$  значення ймовірності вершини  $X^{(j)}$  за умови, що мала місце подія  $E = e$ ,  $P(X^{(i)} | e)$  значення оцінки ймовірності.

#### Контрольні питання

1. Дайте означення ймовірнісного висновку у мережах Байєса. Наведіть приклади.
2. Поясніть своїми словами, що означає термін «ймовірнісний висновок».
3. В чому полягає проблема обчислювальної складності при використанні ймовірнісного висновку в мережах Байєса? Для яких мережевих структур простіше робити обчислення, а для яких складніше?

4. Наведіть методи обчислення якості побудови ймовірнісного висновку в мережах Байєса.
5. Які існують групи методів обчислення ймовірнісного висновку?
6. Які існують методи та алгоритми формування точного ймовірнісного висновку?
7. Наведіть приклади статистичних характеристик для оцінювання якості роботи будь-якого методу побудови ймовірнісного висновку.
8. Які існують методи та алгоритми формування апроксимаційного ймовірнісного висновку?
9. У чому полягає різниця між точним та апроксимаційним висновком?
10. Поясніть суть прямого і зворотного операторів, які використовують при передачі повідомлень.
11. Що означають  $\pi$  – і  $\lambda$  – повідомлення мережі? Для чого вони потрібні?
12. У чому полягає суть алгоритму Перла? Для яких структур він використовується? Вкажіть його переваги та недоліки.
13. Що означає «довіра» стосовно мережі у цілому?
14. Поясніть термін «ініціалізація дерева».
15. У чому полягає алгоритм кластеризації дерева клік (LS-метод)? Для яких структур він застосовується? Вкажіть його переваги та недоліки.
16. Поясніть різницю між термінами: “об’єднані дерева” (junction trees), *дерева клік* (clique trees), *гіпердерева* (hypertrees) та *якісні дерева Маркова* (qualitative Markov trees).
17. Поясніть терміни *моралізування* і *триангуляція* графа.
18. У чому полягає суть алгоритму визначного перетину? Для яких структур він використовується? Проаналізуйте його переваги та недоліки.
19. У чому полягає алгоритм виключення змінних? Для яких структур він застосовується? Проаналізуйте його переваги та недоліки.
20. У чому полягає суть алгоритму символічного ймовірнісного висновку? Для яких структур він використовується? Проаналізуйте його переваги та недоліки.
21. Чим можна пояснити неоднозначність ймовірнісних висновків, отриманих за допомогою різних методів?
22. В чому полягає алгоритм ймовірнісного висновку на основі диференційного підходу? Для яких структур він використовується?
23. Поясніть алгоритм формування точного ймовірнісного висновку на основі навчальних даних. Наведіть приклад його використання.
24. Що означає *нормування значень ймовірностей*?
25. Яким ще може бути тип висновку крім *ймовірнісного*?

## РОЗДІЛ 4 ПОБУДОВА ЙМОВІРНІСНОГО ВИСНОВКУ ІЗ ЗАСТОСУВАННЯМ АЛГОРИТМІВ КЛАСТЕРИЗАЦІЇ

- Суть LS – методу, перший етап LS – методу – побудова об'єднаного дерева
- Другий етап LS-методу – алгоритм розповсюдження
- Приклад побудови дерева клік
- Дослідження результатів роботи методу, одержані при використанні різних програмних продуктів, призначених для побудови мереж Байєса
- Порівняння алгоритмів кластеризації

### 4.1. Архітектури алгоритмів кластеризації

#### *Архітектура Lauritzen-Spiegelhalter (LS)*

Особливістю архітектури LS є те, що спочатку будується зв'язне дерево, так зване об'єднане дерево, а потім розповсюджується в ньому повідомлення. Об'єднане дерево отримують з ациклічного неспрямованого графу шляхом його моралізації, триангуляції та впорядкування клік.

Для перетворення мережі Байєса в моралізований граф виконують процедуру "одруження батьків" шляхом введенням додаткового ребра між кожною парою батьків, потім замінюють спрямований зв'язок між вершинами на неспрямований. Після цього отриманий граф триангулюють, тобто роблять його таким, щоб для будь-якого його простого циклу (цикл, в якому ні одна з вершин не зустрічається двічі) довжиною більше трьох, існувало б ребро, що з'єднує дві не сусідні вершини цього циклу.

Після триангуляції графу ми можемо впорядкувати його кліки (максимально повні підграфи–підграфи, де всі їх вершини суміжні між собою).

Зв'язне дерево – це таке дерево, вершини якого є підмножинами змінних, і змінна, яка міститься у двох різних вершинах, повинна бути в кожній вершині на шляху між цими двома вершинами. Об'єднаним деревом називають таке зв'язне дерево, вершинами якого є кліки триангульованого перед цим графу. Потім вводять між кліками ребра, що містять сепаратори – перетини множин вершин цих клік, видаляють зайві і вибирають корінь дерева.

З кожною клікою пов'язують таблицю умовних ймовірностей, яка містить змінні кліки.

Розповсюдження повідомлення по дереву (пропагація) відбувається за два етапи: сходження догори та донизу.

#### *Правила сходження догори:*

1. Кожен нащадок чекає, щоб надіслати повідомлення своїм батькам до тих пір, доки сам не отримає повідомлення від всіх своїх нащадків. Якщо нащадків у нього немає, то він може відсилати повідомлення одразу.

2. Коли нащадок готовий надсилати повідомлення, він формує його шляхом маргіналізації своєї поточної таблиці по сепаратору – підсумовуванням змінних, що не містяться в сепараторі.

Після надсилання повідомлення батькам, він ділить свою таблицю на це повідомлення (рис. 4.1).

3. Коли батько отримує повідомлення, він множить свою таблицю на нього (рис. 4.1).

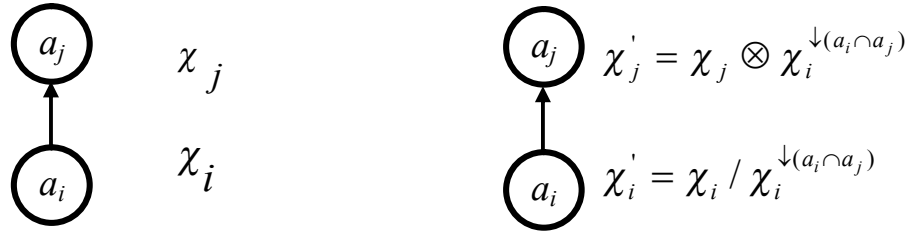


Рис. 4.1. Вершини мережі до та після виконання "сходження догори"

Таблицю  $i$ -ї вершини виражають через функцію  $\chi_i : \Omega_{a_i} \rightarrow R^+$ , де  $\Omega_{a_i}$  - множина можливих станів вершини  $a_i$ . Вираз  $\chi_i^{\downarrow(a_i \cap a_j)}$  означає маргіналізацію таблиці вершини  $a_i$  по сепаратору між нею та вершиною  $a_j$ , а вираз  $\chi_i \otimes \chi_j = \chi_i(x^{\downarrow a_i}) \cdot \chi_j(x^{\downarrow a_j}), \forall x \in \Omega_{a_i \cup a_j}$ .

Сходження догори закінчується, коли корінь дерева отримав повідомлення від усіх своїх нащадків.

*Правила сходження донизу:*

1. Кожен батько чекає, щоб надіслати повідомлення своїм нащадкам, до тих пір, доки сам не отримає повідомлення від своїх батьків. Корінь дерева не має батьків, тому одразу відсилає повідомлення.

2. Коли батько готовий надсилати повідомлення, він формує його шляхом маргіналізації своєї поточної таблиці.

3. Коли нащадок отримує повідомлення, він множить свою таблицю на нього.

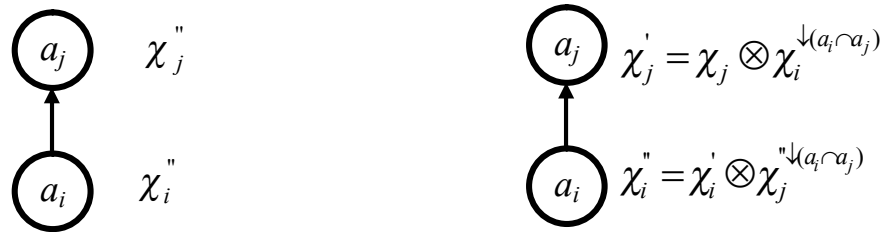


Рис. 4.2. Таблиці вершин до та після сходження донизу

Сходження донизу закінчується, коли всі листи дерева отримали повідомлення від своїх батьків. Після цього з таблиці кожної клітки можна отримати значення її окремих змінних.

### Архітектура Hugin

Нехай є об'єднане дерево і відповідні таблиці для кожної клітки. Сепаратор кожного ребра дерева буде також містити відповідні таблиці. Процес пропagaції відбуватиметься в два етапи: "сходження догори" та "сходження донизу". Особливістю архітектури Hugin є те, що на етапі "сходження догори" відправник не ділить свою таблицю на повідомлення, а замість цього записує його в сепаратор. Це економить зусилля на підрахунки, але й потребує більшого об'єму пам'яті.

На етапі сходження донизу сепаратор розподіляє нове повідомлення на те, яке

він раніше зберігав, і саме на це відношення множить свою таблицю отримувач повідомлення. Економія обчислень відбувається завдяки діленню таблиць сепараторів, так що вони мають менший розмір, ніж таблиці клік.

*Правила пропagaції* [139]:

1. Кожна некоренева вершина чекає, щоб надіслати повідомлення, доки сама не отримає повідомлення.
2. Корінь дерева чекає, щоб надіслати повідомлення, доки не отримає повідомлення від усіх своїх нащадків.

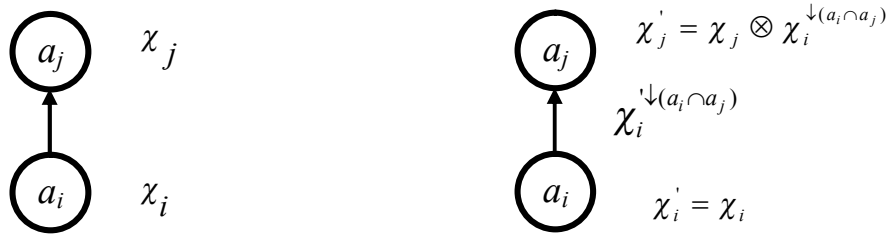


Рис. 4.3. Таблиці вершин до та після "сходження догори"

3. Коли вершина готова надіслати повідомлення, вона формує його шляхом маргіналізації своєї поточної таблиці по сепаратору і тоді надсилає це повідомлення сепаратору між нею та отримувачем.

4. Коли сепаратор отримує "нове" повідомлення від одного зі своїх сусідів, він ділить його на свою поточну, "стару" таблицю, відсилає результат іншому сусідові і потім замінює "старе" значення на "нове".

5. Коли вершина отримує повідомлення, вона множить свою таблицю на нього.

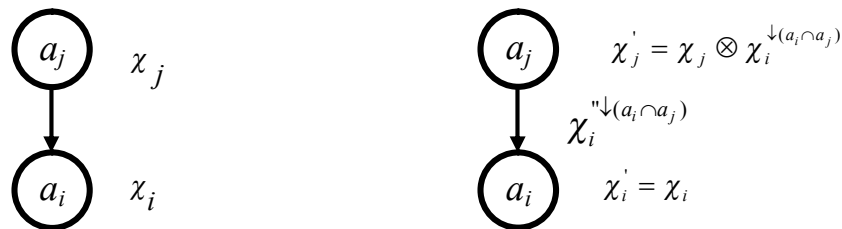


Рис. 4.4. Таблиці вершин до та після "сходження донизу"

Після завершення процесу пропagaції значення окремої змінної можна отримати з таблиці сепаратора, який її містить, раціональніше – найменшого.

Якщо ж жоден сепаратор не містить шукану змінну, то її значення обчислюється з таблиці кліки.

### Архітектура Shenoy-Shafer

В архітектурі Shenoy-Shafer використовується набір таблиць умовних ймовірностей, які визначають сукупний розподіл. Змінні таблиць формують гіперграф, до якого додають підмножини, значення яких необхідно знайти.

На першому кроці формується з підмножин гіперграфу бінарне зв'язне дерево. Бінарне зв'язне дерево – це зв'язне дерево, кожна вершина якого має не більше трьох сусідів. Процес формування спирається на ідею об'єднання [140] та ідею бінарних операцій, тобто усі множення повинні виконуватись попарно.

Наприклад, для мережі Байєса з трьох змінних, позначених як  $D$  – хвороба,  $S_1$  та  $S_2$  – її симптоми, двійкове зв'язне дерево буде мати вигляд, представлений на рис. 4.5.



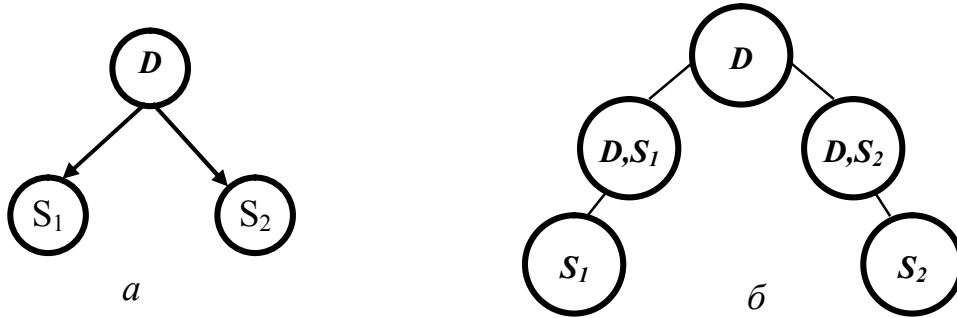


Рис. 4.5. Мережа Байєса (а), двійкове зв'язне дерево (б)

З кожною вершиною дерева пов'язується відповідна таблиця. Вершина, значення ймовірності якої потрібно знайти, запитує повідомлення від своїх сусідів.

*Реалізація правил пропагації:*

1. *Розрахунок повідомлення.* Припустимо, що  $r$  і  $s$  сусіди і  $s$  надсилає запит на повідомлення до  $r$ .  $r$ , в свою чергу, надсилає запити на повідомлення від своїх інших сусідів і після отримання цих повідомлень, від розраховує повідомлення  $s$  таким чином. Фактично, повідомлення, які  $r$  надсилає своєму сусіду  $s$ , є комбінацією усіх отриманих ним повідомлень від інших сусідів, маргіналізованих по  $r \cap S$ . Позначимо повідомлення як  $\mu^{r \rightarrow s}$ , сусідів  $r$  у дереві – як  $N(r)$ , а таблицю  $r$  як  $\chi_r$ . Тоді  $\mu^{r \rightarrow s} = (\otimes \{ \mu^{t \rightarrow r} \mid t \in (N(r) - \{s\}) \}) \otimes \chi_r$ . Лист у дереві має тільки одного сусіда, тому коли йому надходить запит на повідомлення, від може одразу його відіслати.

2. *Розрахунок шуканого значення вершини.* Коли до вершини  $s$  надійшли повідомлення від усіх сусідів, комбінуючи їх з її власною таблицею  $\chi_s$ , можна знайти шукане значення так:

$$\varphi^{\downarrow s} = \otimes \{ \mu^{t \rightarrow s} \mid t \in (N(s)) \} \otimes \chi_s, \text{ де } \varphi - \text{таблиця сукупного розподілу.}$$

Кожна вершина у двійковому зв'язному дереві може зберігати нуль, одну, дві або більше таблиць – одну для кожної вхідної таблиці і одну для вихідної (якщо шукається її значення). Кожен сепаратор може зберігати щонайбільше дві таблиці для двох повідомлень в обидва напрямки. На рис. 4.6 зображені таблиці для простого дерева з двох вершин.

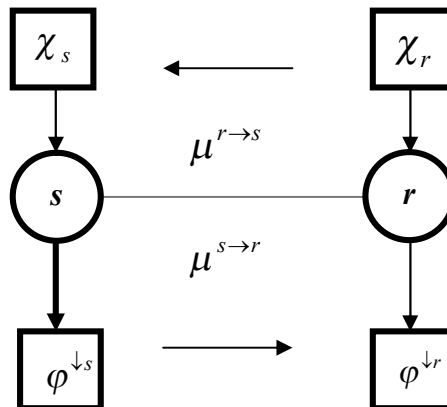


Рис. 4.6. Таблиці для двійкового зв'язного дерева

Обидві вершини зберігають по одній вхідній таблиці, значення обох вершин шукані, тому вони також зберігають по одній вихідній таблиці.

Отже, при реалізації ймовірнісного висновку відповідно до алгоритмів кластеризації, необхідно спочатку привести структуру мережі Байєса до вигляду зв'язного дерева (joint tree), зв'язати з вершинами цього дерева таблиці умовних ймовірностей первинної структури мережі, а потім використовувати алгоритми пропagaції, що ґрунтуються на проходженні повідомлень по дереву в обох напрямках та послідовному перераховуванню таблиць умовних ймовірностей вершин дерева.

#### 4.2. Застосування *LS* – методу та його модифікацій

Ідея *LS*-методу (Lauritzen-Spiegelhalter) формування ймовірнісного висновку є основоположною ідеєю методів кластеризації, при використанні яких для реалізації ймовірнісного висновку необхідно спочатку привести структуру мережі Байєса до вигляду об'єднаного дерева, а потім використовувати алгоритм розповсюдження повідомлень по дереву догори та донизу і послідовно перераховувати таблиці умовних ймовірностей вершин дерева.

У загальному вигляді *LS*-метод передбачає виконання двох етапів. На першому етапі виконується побудова об'єднаного дерева клік з первинної структури мережі та заповнення вершин цього дерева таблицями умовних ймовірностей мережі. На другому етапі обчислюють значення ймовірностей станів вершин на основі алгоритмів розповсюдження значень ймовірності по об'єднаному дереву. На базі *LS*-методу створено не один метод обчислення ймовірнісного висновку.

Як приклад побудови ймовірнісного висновку розглянемо завдання оцінювання кредитоспроможності фізичних осіб при отриманні споживчого кредиту. Для побудови скорингової моделі використовується база даних клієнтів комерційного банку. База даних складається з 3347 записів про клієнтів.

В якості змінних процесу використані такі атрибути: стать (*G* – gender); вік (*A* – age); сімейний стан (*M* – marital status); тип трудової зайнятості (*T* – type of contract); поручитель (*P* – contact person); сума кредиту (*C* – total credit amount); результат: кредитоспроможний? (*R* – result (credit worthiness)).

Для побудови топології мережі використано евристичний метод побудови мережі Байєса за навчальними даними [141]. На рис. 4.7 зображена структура мережі Байєса.

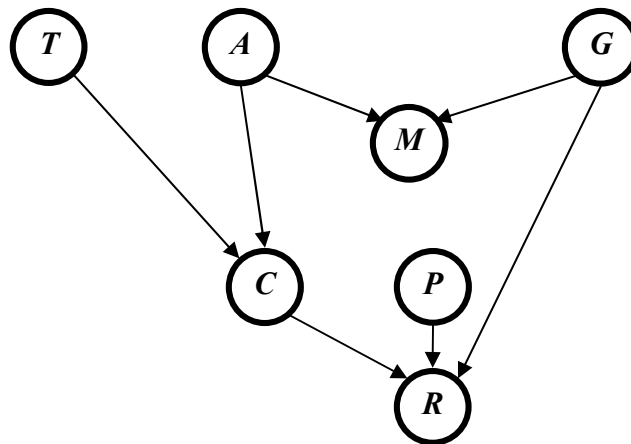


Рис. 4.7. Структура мережі Байєса для оцінювання кредитоспроможності

Далі для кожної вершини визначаються умовні ймовірності. Результати представлені в таблицях 4.1 – 4.7.

Таблиця 4.1

Значення умовної ймовірності вершини T

Стан	Ймовірність
T1 = Employed full time	0,875
T2 = Pensioner	0,061
T3 = Self Employer	0,064

Таблиця 4.2

Значення умовної ймовірності вершини A

Стан	Ймовірність
A1=More then 36	0,45
A2=Low then 36	0,55

Таблиця 4.3

Значення умовної ймовірності вершини G

Стан	Ймовірність
F=Female	0,47
M=Male	0,53

Таблиця 4.4

Значення умовної ймовірності вершини P

Стан	Ймовірність
No	0,44
Yes	0,56

Таблиця 4.5

Значення умовної ймовірності вершини M

Батьки	Стани батьків			
	A1	A2	A1	A2
A				
G	F	F	M	M
Стан	Ймовірність			
M1 = Civil marriage	0,0361	0,0841	0,0321	0,0903
M2 = Divorced	0,1996	0,0771	0,0681	0,0819
M3 = Married	0,5848	0,5455	0,8517	0,1218
M4 = Single	0,0792	0,2853	0,0341	0,706
M5 = Widowed	0,1003	0,008	0,014	0

Таблиця 4.6

Значення умовної ймовірності вершини C

Батьки	Стани батьків					
T	T1	T1	T2	T2	T3	T3
A	A1	A2	A1	A2	A1	A2
Стан	Ймовірність					
C1= Lower then 2100 UAH	0,55	0,6	0,63	0,8	0,4	0,46
C2= More then 2100 UAH	0,45	0,4	0,37	0,2	0,6	0,54

Таблиця 4.7

Значення умовної ймовірності вершини R

Батьки	Стани батьків							
G	F	F	F	F	M	M	M	M
P	No	No	Yes	Yes	No	No	Yes	Yes
C	C1	C2	C1	C2	C1	C2	C1	C2
Стан	Ймовірність							
R1=Good	0,96	0,91	0,99	0,99	0,86	0,76	0,985	0,9659
R2=Bad	0,04	0,09	0,01	0,01	0,14	0,24	0,015	0,0341

Далі для побудови структури, необхідної для формування ймовірнісного висновку, необхідно послідовно виконати такі етапи.

**Етап 1. Моралізація графа:** послідовно перебираються усі вершини мережі Байєса, у яких є батьки. Якщо батьківські вершини не зв'язані між собою, то між ними вводиться зв'язок ("сусід"). У результаті одержано моралізований граф, представлений на рис. 4.8.

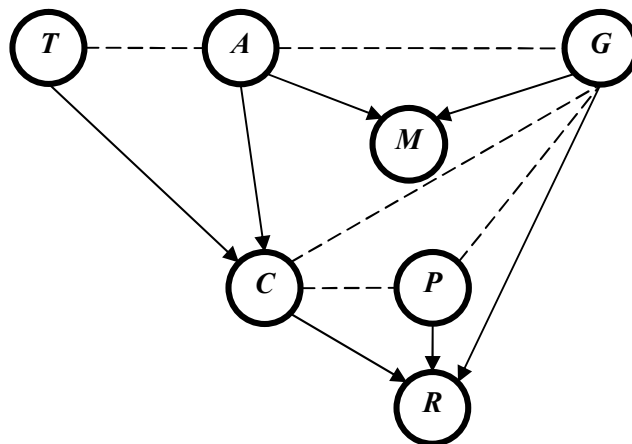


Рис. 4.8. Моралізований граф

**Етап 2. Приведення графа до неспрямованої форми:** для всіх вершин зв'язок "батько-нащадок" замінюємо на зв'язок "сусід". У результаті перетворень одержано граф, представлений на рис. 4.9.

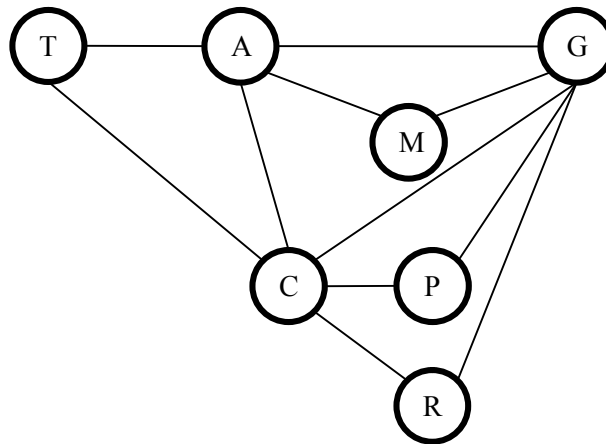


Рис. 4.9. Неспрямований граф

**Етап 3. Триангуляція графа** реалізується за два кроки. При виконанні **кроку 1** послідовно перебираються усі вершини мережі Байєса до тих пір, доки не будуть розглянуті усі вершини.

Спочатку перевіряємо, чи є суміжними між собою сусіди аналізованої вершини. Якщо так, то така вершина симпліціальна, тобто утворює кліку разом із своїми сусідами. Така вершина виключається із розгляду разом з її ребрами. Виділимо симпліціальні вершини  $T$  і  $M$  і виключимо їх та їхні ребра (рис. 4.10).

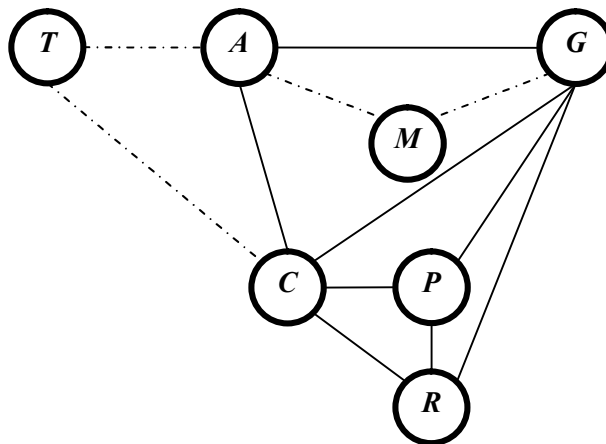


Рис. 4.10. Триангуляція графа (виключення вершин  $T$  і  $M$ )

Вершини  $T$  і  $M$  мають суміжних між собою сусідів, такі вершини є симпліціальними, тобто утворюють кліку разом із своїми сусідами. Тому їх слід виключити із розгляду разом з її ребрами.

Далі виділяємо вершину  $A$ . Перевіряємо, чи є суміжними між собою сусіди аналізованої вершини.

Оскільки сусіди вершини  $A$  є суміжними між собою, то така вершина є симпліціальною і її також слід виключити із розгляду разом з ребрами.

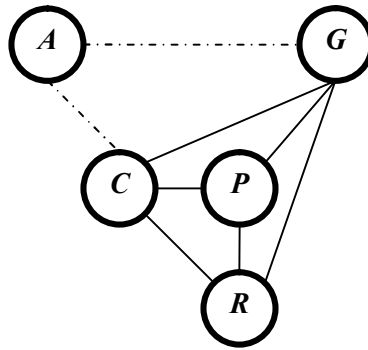


Рис. 4.11. Триангуляція графа (виключення вершини A)

Далі перебираємо інші вершини і якщо після перебору всіх вершин мережа, що залишилася до розгляду, не порожня, то виконуємо крок 2 етапу триангуляції. Якщо ж вершин з найбільшою кількістю сусідів не залишилося, то граф триангульовано. Оскільки у графі, що залишився, усі вершини суміжні між собою, то хоча він і складається з чотирьох вершин, але є повним. Тому він буде представлений однією клікою, а не трьома.

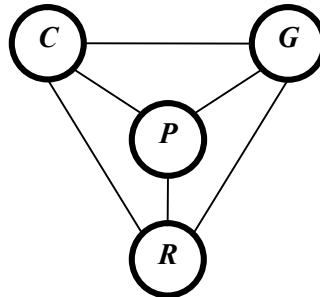


Рис. 4.12. Триангульований граф

Таким чином, необхідності введення додаткових ребер немає, початковий граф триангульовано. Тобто *крок 2* етапу триангуляції графу виконувати не потрібно

**Етап 4. При побудові дерева клік з сепараторами** необхідно послідовно перебрати усі кліки, отримані на етапі триангуляції. Зокрема перевірити, чи є поточна кліка підмножиною інших не перебраних клік. Якщо так, то вона знищується. У випадку, коли в поточній та інших не перебраних кліках співпадає хоча б одна вершина, то між відповідними кліками вводиться ребро, що містить сепаратор – перетин множин вершин цих клік. У результаті виконання дій даного етапу буде побудоване дерево клік, представлене на рис. 4.13.

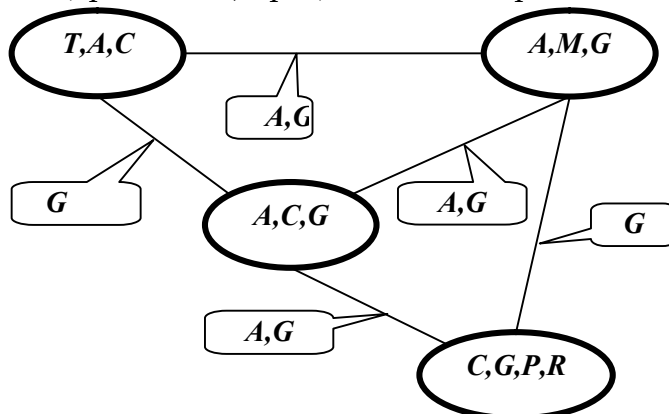


Рис. 4.13. Побудоване дерево клік

На етапі 5 виконується **побудова об'єданого дерева**. Всі кліки послідовно зв'язують між собою шляхом вибору для зв'язку ребер з найбільшими сепараторами, а інші ребра видаляються. Дерево, що зв'яже всі кліки з максимальними потужностями сепараторів, буде об'єднаним деревом.

Для зручності за корінь у об'єданому дереві вибирається кліка з найбільшою кількістю вершин. Якщо таких вершин декілька, перевага віддається кліці з найбільшою кількістю ребер. Побудоване дерево представлено на рис. 4.14.

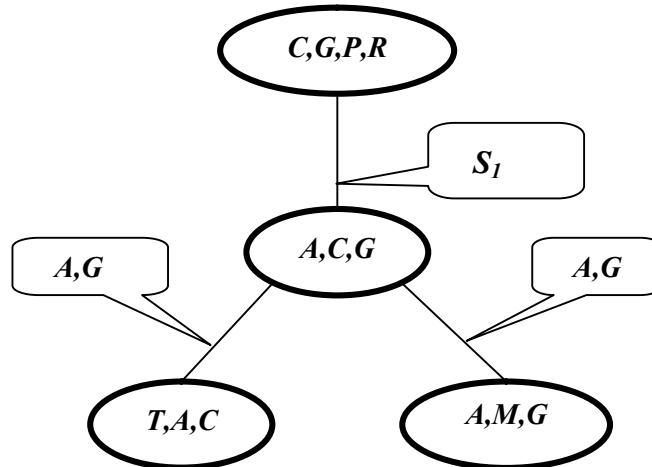


Рис. 4.14. Побудоване об'єдане дерево

Наступним етапом (**етап 6**) є **заповнення об'єданого дерева таблицями**. Розглядаються "невідмічені" вершини в необробленій кліці, для чого послідовно виконуються такі дії.

**Дія 1.** Якщо серед "невідмічених" вершин є така, яка не зустрічається в інших необроблених кліках, то таблиця повинна мати вигляд  $P(\text{this node} | \text{other nodes in clique})$  – у такому разі ця вершина "відмічена", і кліка вважається обробленою.

**Дія 2.** Якщо у первинній мережі Байєса такої таблиці немає або така вершина не одна, то використовується таблиця сукупного розподілу ймовірності – у такому випадку усі вершини кліки "відмічені", і кліка оброблена.

**Дія 3.** Якщо в кліці є вже "відмічені" вершини і залишилась всього одна "невідмічена" вершина, то використовується таблиця вигляду  $P(\text{unmarket node})$  з первинної мережі Байєса – у такому разі ця вершина "відмічена", і кліка вважається обробленою.

**Дія 4.** Якщо у кліці вже є "відмічені" вершини, а "невідмічених" вершин декілька, то використовується таблиця сукупного розподілу ймовірності "невідмічених" вершин, і тоді ці вершини "відмічені", а кліка вважається обробленою.

**Дія 5.** Якщо в необробленій кліці усі вершини вже "відмічені", то кліка заповнюється таблицею виду  $P(\text{any node})$  і вважається обробленою.

Процес починається з листів дерева, потім послідовно перебираються усі кліки-листя та корінь дерева. Як видно з побудованого об'єданого дерева (рис. 4.15), в даній задачі розглядаються три кліки-листя (кліка-лист 1 – "T, A, C", кліка-лист 2 – "A, M, G", кліка-лист 3 – "A, C, G") та корінь дерева.

У кліці-листі 1 - "T, A, C" знаходиться одна "невідмічена" вершина, яка більше не зустрічається в інших кліках – T. Таблиці вигляду  $P(T | C, A)$  не існує у первинній структурі мережі Байєса, тому використовується таблиця сукупного розподілу ймовірностей, а вершини T, A, C – "відмічені".

Таблиця 4.8

Значення кліки-листа 1 – "T, A, C"

$C=C1$	$C=C2$	T	A
0,216563	0,177188	T1	A1
0,28875	0,1925	T1	A2
0,017294	0,010157	T2	A1
0,02684	0,00671	T2	A2
0,01152	0,01728	T3	A1
0,016192	0,019008	T3	A2

Кліка-лист 2 – "A, M, G". У цій кліці знаходиться одна вершина, яка більше не зустрічається в інших кліках – M. Таблиця вигляду  $P(M | A, G)$  існує і слід використовувати її. Вершина M вважається "відміченою".

Таблиця 4.9

Значення кліки-листа 2 - "A, M, G"

M=M1	M=M2	M=M3	M=M4	M=M5	A	G
0,0361	0,1996	0,5848	0,0792	0,1003	A1	F
0,0321	0,0681	0,8517	0,0341	0,014	A1	M
0,0841	0,0771	0,5455	0,2853	0,008	A2	F
0,0903	0,0819	0,1218	0,706	0	A2	M

Кліка-лист 3 – "A, C, G". У цій кліці знаходяться дві "відмічені" вершини – A, C і одна "невідмічена" – G, тому використовується таблиця вигляду  $P(G)$ . Вершина G "відмічена".

Таблиця 4.10

Значення кліки-листа 3 - "A, C, G"

$C=C1$	$C=C2$	A	G
0,47	0,47	A1	F
0,53	0,53	A1	M
0,47	0,47	A2	F
0,53	0,53	A2	M

Корінь дерева – "R, C, P, G". У цій кліці є дві "невідмічені" вершини, яких немає в необроблених кліках – P, R, а тому використовується таблиця сукупного розподілу ймовірностей – тепер усі вершини кліки "відмічені".



Таблиця 4.11

Таблиця сукупного розподілу ймовірностей для кореня дерева – "R, C, P, G"

<i>R=R1</i>	<i>R=R2</i>	<i>C</i>	<i>P</i>	<i>G</i>
0,114551	0,004772944	<i>C1</i>	No	<i>F</i>
0,115719	0,018837896	<i>C1</i>	No	<i>M</i>
0,150348	0,001518664	<i>C1</i>	Yes	<i>F</i>
0,168685	0,002568804	<i>C1</i>	Yes	<i>M</i>
0,079604	0,007872876	<i>C2</i>	No	<i>F</i>
0,074969	0,023674464	<i>C2</i>	No	<i>M</i>
0,11022	0,001113336	<i>C2</i>	Yes	<i>F</i>
0,121265	0,004281132	<i>C2</i>	Yes	<i>M</i>

*Другий етап LS-методу (алгоритм розповсюдження)* умовно поділяється на три підетапи.

Для кожного спостереження змінної вибирається одна таблиця, яка містить цю змінну. Задаємо нульовими усі входження, які суперечать спостереженню. За змістом задачі, що розглядається – оцінювання кредитоспроможності фізичних осіб при отриманні споживчого кредиту, розглянемо випадок, коли інстанційовано вершини  $P=No$ ,  $G=M$ . Для спостереження  $P=No$  заповнимо нулями входження  $P=Yes$  в корені об'єднаного дерева (табл. 4.12).

Таблиця 4.12

Значення кореня дерева – "R, C, P, G"

<i>R = R1</i>	<i>R = R2</i>	<i>C</i>	<i>P</i>	<i>G</i>
0,114551	0,004772944	<i>C1</i>	No	<i>F</i>
0,115719	0,018837896	<i>C1</i>	No	<i>M</i>
0	0	<i>C1</i>	Yes	<i>F</i>
0	0	<i>C1</i>	Yes	<i>M</i>
0,079604	0,007872876	<i>C2</i>	No	<i>F</i>
0,074969	0,023674464	<i>C2</i>	No	<i>M</i>
0	0	<i>C2</i>	Yes	<i>F</i>
0	0	<i>C2</i>	Yes	<i>M</i>

Для спостереження  $G = M$ , нулями заповнимо входження  $G = F$  в кліці 3 – "A, C, G", табл. 4.13.

Таблиця 4.13

Значення кліки-листа 3 – "A, C, G"

<i>C=C1</i>	<i>C=C2</i>	<i>A</i>	<i>G</i>
0	0	<i>A1</i>	<i>F</i>
0,53	0,53	<i>A1</i>	<i>M</i>
0	0	<i>A2</i>	<i>F</i>
0,53	0,53	<i>A2</i>	<i>M</i>

*Підетап 1. Сходження догори* передбачає, що кожен лист дерева посилає повідомлення своїй батьківській змінній. Повідомлення є результатом маргіналізації – підсумовування змінних таблиці, які не містяться в сепараторі. Після відсилання повідомлення відправник ділить свою поточну таблицю умовних ймовірностей на

нього. Коли отримувачу надходить повідомлення, він множить його на свою таблицю умовних ймовірностей і таким чином виходить нова таблиця. Коли отримувачу надходять повідомлення від усіх його нащадків, то він, у свою чергу, посилає також повідомлення своєму батьку і ділить свою таблицю на відіслане повідомлення. Процес триває до тих пір, доки корінь зв'язного дерева не отримає повідомлення від усіх своїх нащадків [142].

Таблиця 4.14

Передача повідомлення від кліки 1 кліці 3 (маргіналізація по A, C)

$C=C1$	$C=C2$	A
0,245376	0,204624	A1
0,331782	0,218218	A2

Таблиця 4.15

Нова таблиця кліки 1

$C=C1$	$C=C2$	T	A
0,882574	0,865917	T1	A1
0,8703	0,882145	T1	A2
0,070478	0,049635	T2	A1
0,080896	0,030749	T2	A2
0,046948	0,084448	T3	A1
0,048803	0,087106	T3	A2

Таблиця 4.16

Передача повідомлення від кліки 2 кліці 3 (маргіналізація по A, G)

	A	G
1	A1	F
1	A1	M
1	A2	F
1	A2	M

Таблиця 4.17

Нова таблиця кліки 2

$M=M1$	$M=M2$	$M=M3$	$M=M4$	$M=M5$	A	G
0,0361	0,1996	0,5848	0,0792	0,1003	A1	F
0,0321	0,0681	0,8517	0,0341	0,014	A1	M
0,0841	0,0771	0,5455	0,2853	0,008	A2	F
0,0903	0,0819	0,1218	0,706	0	A2	M

Таблиця 4.18

Нова таблиця кліки 3

$C=C1$	$C=C2$	A	G
0	0	A1	F
0,130049	0,10845072	A1	M
0	0	A2	F
0,175844	0,11565554	A2	M

Таблиця 4.19

Повідомлення від кліки з кореню (маргіналізація по  $C, G$ )

$C=C1$	$C=C2$	$G$
0	0	$F$
0,305894	0,22410626	$M$

Таблиця 4.20

Нова таблиця кліки з

$C=C1$	$C=C2$	$A$	$G$
0	0	$A1$	$F$
0,4251453	0,483925438	$A1$	$M$
0	0	$A2$	$F$
0,5748547	0,516074562	$A2$	$M$

Таблиця 4.21

Нова таблиця кореня

$R=R1$	$R=R2$	$C$	$P$	$G$
0	0	$C1$	No	$F$
0,035398	0,005762394	$C1$	No	$M$
0	0	$C1$	Yes	$F$
0	0	$C1$	Yes	$M$
0	0	$C2$	No	$F$
0,016801	0,005305596	$C2$	No	$M$
0	0	$C2$	Yes	$F$
0	0	$C2$	Yes	$M$

**Підетап 2 (сходження донизу).** Корінь посилає повідомлення кожному своєму нащадкові. Він ділить свою таблицю умовних ймовірностей на повідомлення, отримане від нащадка, маргіналізує таблицю по сепаратору і посилає результат. Коли нащадок отримує повідомлення від свого батька, він множить його на свою поточну таблицю умовних ймовірностей і формує таким чином нову таблицю. Далі він маргіналізує її (по сепаратору) і посилає своєму нащадкові.

Таблиця 4.22

Повідомлення від кореня кліці з (маргіналізація по  $C, G$ )

$C = C1$	$C = C2$	$G$
0,1193236	0,0874764	$F$
0,1345564	0,0986436	$M$

Таблиця 4.23

Нова таблиця кліки з:

$C = C1$	$C = C2$	$A$	$G$
0	0	$A1$	$F$
0,057206018	0,047736	$A1$	$M$
0	0	$A2$	$F$
0,077350382	0,050907	$A2$	$M$

Таблиця 4.24

Повідомлення від кліки 3 кліці 2 (маргіналізація по A, G)

	A	G
0	A1	F
0,104942	A1	M
0	A2	F
0,128258	A2	M

Таблиця 4.25

Нова таблиця кліки 2

M = M1	M = M2	M = M3	M = M4	M = M5	A	G
0	0	0	0	0	A1	F
0,003368644	0,007147	0,089379243	0,003579	0,001469	A1	M
0	0	0	0	0	A2	F
0,011581682	0,010504	0,015621804	0,09055	0	A2	M

Таблиця 4.26

Повідомлення від кліки 3 кліці 1 (маргіналізація по A, C)

C = C1	C = C2	A
0,057206	0,047736147	A1
0,0773504	0,050907453	A2

Таблиця 4.27

Нова таблиця кліки 1

C = C1	C = C2	T	A
0,05048855	0,041335565	T1	A1
0,067318066	0,044907774	T1	A2
0,00403174	0,002369381	T2	A1
0,006257375	0,001565357	T2	A2
0,002685729	0,004031202	T3	A1
0,003774941	0,004434322	T3	A2

Процес маргіналізації триває до тих пір, доки усі листки не отримають повідомлення. Таким чином, результатом будуть знову перераховані таблиці для кожної змінної за умови наявності спостережень, які потім необхідно нормалізувати. Отже, в результаті отримано нормалізовані таблиці.

Таблиця 4.28

Кліка-лист 1 – "T, A, C"

C = C1	C = C2	T	A
0,225267	0,112678	T1	A1
0,406123	0,130548	T1	A2
0,017989	0,006459	T2	A1
0,03775	0,004551	T2	A2
0,011983	0,010989	T3	A1
0,022774	0,012891	T3	A2

Таблиця 4.29

Кліка-лист 2 – "A, M, G"

$M = M1$	$M = M2$	$M = M3$	$M = M4$	$M = M5$	A	G
0	0	0	0	0	A1	F
0,014306465	0,030351099	0,379589294	0,015197834	0,00624	A1	M
0	0	0	0	0	A2	F
0,05005471	0,045398458	0,067515656	0,391346905	0	A2	M

Таблиця 4.30

Кліка 3 – "A, C, G"

$C = C1$	$C = C2$	A	G
0	0	A1	F
0,276591	0,169093	A1	M
0	0	A2	F
0,373989	0,180327	A2	M

Таблиця 4.31

Корінь дерева – "R, C, P, G"

$R = R1$	$R = R2$	C	P	G
0	0	C1	No	F
0,559498	0,091081	C1	No	M
0	0	C1	Yes	F
0	0	C1	Yes	M
0	0	C2	No	F
0,26556	0,083861	C2	No	M
0	0	C2	Yes	F
0	0	C2	Yes	M

**Підетап 3. (Розрахунок ймовірностей станів вершин).** Для кожної вершини виконується пошук клік, у яких вона міститься. Якщо серед них є кліка, яка не є листом, то вибирається ця кліка. Інакше – будь-яка з них.

Для знаходження ймовірності кожного стану вершини необхідно підсумувати усі значення ймовірностей для цього конкретного стану, які є в таблиці кліки.

Нижче представлено процес обчислення значень ймовірностей станів вершин мережі Байєса для задачі кредитного скорингу у випадку, коли є дві інстанційовані вершини  $P = No$  і  $G = M$ .

Для вершини  $T$  підсумовуються її значення в кліці 1:

$$P(T = T1) = 0,225267 + 0,112678 + 0,406123 + 0,130548 = 0,874616$$

$$P(T = T2) = 0,017989 + 0,006459 + 0,03775 + 0,004551 = 0,066749$$

$$P(T = T3) = 0,011983 + 0,010989 + 0,022774 + 0,012891 = 0,058637$$

Для вершини  $A$  підсумовуються її значення в кліці 3:

$$P(A = A1) = 0 + 0 + 0,276591 + 0,169093 = 0,445684$$

$$P(A = A2) = 0 + 0 + 0,373989 + 0,180327 = 0,554316$$

Для вершини  $M$  підсумовуються її значення в кліці 2:

$$P(M = M1) = 0 + 0,014306465 + 0 + 0,05005471 = 0,064361175$$

$$P(M = M2) = 0 + 0,030351099 + 0 + 0,045398458 = 0,075749557$$

$$P(M = M3) = 0 + 0,379589294 + 0 + 0,067515656 = 0,44710495$$

$$P(M = M4) = 0 + 0,015197834 + 0 + 0,391346905 = 0,406544739$$

$$P(M = M5) = 0 + 0,00624 + 0 + 0 = 0,00624$$

Для вершини  $C$  значення підсумовуються в кліщі  $Z$ :

$$P(C = C1) = 0 + 0,276591 + 0 + 0,373989 = 0,65058$$

$$P(C = C2) = 0 + 0,169093 + 0 + 0,180327 = 0,34942$$

Для вершини  $R$  значення підсумовуються в корені :

$$P(R = R1) = 0 + 0,559498 + 0 + 0 + 0,26556 + 0 + 0 = 0,825058$$

$$P(R = R2) = 0 + 0,091081 + 0 + 0 + 0,083861 + 0 + 0 = 0,174942$$

Отже, задача кредитного скорингу є достатньо ресурсо- та трудомісткою, тому для її розв'язку існує ціла низка різноманітних програмних продуктів, призначених для роботи з мережами Байєса, які суттєво відрізняються за продуктивністю, швидкістю, обсягами використовуваної пам'яті. Тому доцільно дослідити їх роботу, а саме використовуючи в якості тестових наборів дані щодо кредитоспроможності фізичних осіб.

#### **4.3. Дослідження результатів роботи LS-методу та його модифікацій, одержаних при використанні різних програмних продуктів, призначених для побудови мереж Байєса**

Мережі Байєса стають дедалі популярнішим інструментом різноманітних експертних систем, широко застосовуються у підсистемах інтелектуального аналізу даних систем підтримки прийняття рішень. Особливо актуальне їх застосування при розв'язанні задач, що характеризуються наявністю невизначеностей різного роду та великими обсягами вхідних даних, необхідністю пошуку причинно-наслідкових взаємозв'язків та моделюванні різноманітних процесів на основі використання великих статистичних баз даних.

Тому розробники програмного забезпечення систем підтримки прийняття рішень приділяють даному питанню багато уваги. Існує значна кількість різноманітних програмних продуктів, прикладами яких можна назвати Netica, MSBNx, GeNie, Hugin. Вартість бізнес-версій даних програмних засобів коливається в межах від 140 до 5200 €, в залежності від типу ліцензії та комплектації.

Програма Netica розроблена канадською компанією Norsys Software Corporation. Для побудови ймовірнісного висновку в ній реалізовано метод виключення зв'язних дерев (elimination junction tree method).

MSBNx – розробка фірми MicroSoft, для побудови ймовірнісного висновку в ній використовується метод, заснований на поширенні повідомлень по дереві клік.

GeNie – безкоштовна програма, розроблена в університеті Піттсбурга. Перевагами даного продукту є зручність інтерфейсу, наявність найбільш популярних алгоритмів для роботи з мережею Байєса, включаючи кластерне і стохастичне моделювання, а крім того, можливість під'єднання додатково створених власних модулів з алгоритмами, реалізованими мовою програмування C.

Програма Hugin Expert розроблена датською компанією Hugin Expert A/S. Для побудови ймовірнісного висновку застосовується модифікація LS-методу.

У програмному забезпеченні BayesNet, розробленому співробітниками кафедри "Математичних методів системного аналізу" НТУУ "КПІ", використовується метод ймовірнісного висновку на основі навчальних даних. [6]. Дане програмне забезпечення є безкоштовним, його можна знайти за посиланням <https://sites.google.com/site/data4mining/bnetmaster>. Результати роботи даної розробки є прийнятними, що підтверджено тестуванням на реальних наборах даних.

Всі вказані програмні продукти, орієнтовані в більшій чи меншій мірі на вирішення вузькоспеціалізованих задач, мають індивідуальні особливості, тому результати побудови ймовірнісного висновку можуть відрізнятись. Тому, досліджуючи ефективність роботи LS-алгоритму та його модифікацій, було порівняно результати розрахунків значень станів вершин мережі Байєса за даним алгоритмом та програмними засобами Netica, MSBNx, GeNIe, Hugin та BayesNet.

В якості змінних процесу використані такі атрибути: стать ( $G$  – gender); вік ( $A$  – age); сімейний стан ( $M$  – marital status); тип трудової зайнятості ( $T$  – Type of contract); поручитель ( $P$  – contact person); сума кредиту ( $C$  – total credit amount); результат: кредитоспроможний? ( $R$  – result (credit worthiness)). Результати обчислень станів вершин мережі Байєса для задачі кредитного скорингу, отримані за допомогою різних програмних продуктів у разі інстанціювання вершин  $P = No$  (немає поручителя) і  $G = M$  (стать – чоловіча), представлені в табл. 4.32.

Таблиця 4.32

Значення станів вершин мережі Байєса, отримані за допомогою різних програмних продуктів для моделі кредитного скорингу у випадку інстанціювання вершин  $P = No$  і  $G = M$

Вершина	Стан	Значення станів вершин, отримані за допомогою програмних продуктів:					
		LS-метод	BayesNet	Netica	MSBNx	GeNIe	Hugin
T	T1	0,8749	0,875	0,875	0,875	0,875	0,875
	T2	0,0609	0,061	0,061	0,061	0,061	0,061
	T3	0,064	0,064	0,064	0,064	0,064	0,064
A	A1	0,45	0,45	0,45	0,45	0,45	0,45
	A2	0,5499	0,55	0,55	0,55	0,55	0,55
G	F	0	0	0	0	0	0
	M	1	1	1	1	1	1
M	M1	0,0641	0,0641	0,0641	0,0641	0,06411	0,0641
	M2	0,0757	0,0757	0,0757	0,0757	0,07569	0,0757
	M3	0,4503	0,4503	0,45	0,4503	0,450255	0,4503
	M4	0,4036	0,4036	0,404	0,4036	0,403645	0,4036
	M5	0,0063	0,0063	0,0063	0,0063	0,0063	0,0063
C	C1	0,577	0,5772	0,577	0,5772	0,577158	0,5793
	C2	0,423	0,4228	0,423	0,4228	0,422842	0,4207
P	No	1	1	1	1	1	1
	Yes	0	0	0	0	0	0
R	R1	0,825	0,8177	0,818	0,8177	0,8177158	0,8179
	R2	0,175	0,1823	0,182	0,1823	0,1822842	0,1821

Як видно з таблиці 4.32, значення станів вершин, отриманих за допомогою різних програмних засобів та LS-методом, практично збігаються. Для більшості вершин значення станів, отриманих у програмних продуктах BayesNet та Netica збігаються, а найбільше розбіжностей з результатами, одержаними у Hugin. Поглиблений аналіз результатів, представлених в таблиці 4.32, дозволив зробити висновок про те, що клієнти чоловічої статі без поручителя більше схильні до неповернення кредитів в 17,5% випадків, в той час як загальний рівень дефолтів по вибірці становить 5,3%.

Таблиця 4.33

Результати обчислень станів вершин мережі Байєса різними програмами для задачі кредитного скорингу у випадку, коли інстанційованих вершин немає, тобто без спостережень

Вершина	Стан	Результати обчислень станів вершин мережі Байєса програмами					
		LS-метод	BayesNet	Netica	MSBNx	GeNie	Hugin
T	T1	0,875	0,875	0,875	0,875	0,875	0,875
	T2	0,061	0,061	0,061	0,061	0,061	0,061
	T3	0,064	0,064	0,064	0,064	0,064	0,064
A	A1	0,45	0,45	0,45	0,45	0,45	0,45
	A2	0,55	0,55	0,55	0,55	0,55	0,55
G	F	0,47	0,47	0,47	0,47	0,47	0,47
	M	0,53	0,53	0,53	0,53	0,53	0,53
M	M1	0,0634	0,0634	0,0634	0,0634	0,0633533	0,0634
	M2	0,1023	0,1023	0,102	0,1023	0,1022614	0,1023
	M3	0,5033	0,5033	0,503	0,5033	0,5033321	0,5033
	M4	0,3044	0,3044	0,304	0,3044	0,3044327	0,3044
	M5	0,0266	0,0266	0,0266	0,0266	0,0266204	0,0266
C	C1	0,577	0,5772	0,577	0,5772	0,577158	0,5793
	C2	0,423	0,4228	0,423	0,4228	0,422842	0,4207
P	No	0,44	0,44	0,44	0,44	0,44	0,44
	Yes	0,56	0,56	0,56	0,56	0,56	0,56
R	R1	0,9365	0,9354	0,935	0,9354	0,9353661	0,9355
	R2	0,0635	0,0646	0,065	0,0646	0,0646339	0,0645



Результати, представлені в таблиці 4.33 вказують на те, що при використанні мережі Байєса ймовірність того, що клієнт не поверне кредит, дорівнює близько 6,4%, якщо про нього відсутня будь-яка апіорна інформація.

У табл. 4.34-4.37 наведено значення ймовірностей вершин мережі Байєса, отриманих при моделюванні різних ситуацій, пов'язаних із потенційними можливостями клієнтів банку повернути одержані кредити.

У табл. 4.34 представлені результати обчислень для випадку коли клієнт не кредитоспроможний -  $R = R2$  (табл. 4.34)

Таблиця 4.34

Результати обчислень станів вершин мережі Байєса різними програмами для задачі кредитного скорингу у випадку інстанційовання вершини  $R = R2$

Вершина	Стан	Значення ймовірності, обчислені у програмах					
		LS-метод	BayesNet	Netica	MSBNx	GeNIe	Hugin
T	T1	0,8748	0,8748	0,875	0,8748	0,8748179	0,8747
	T2	0,0556	0,0556	0,0556	0,0556	0,0555681	0,0556
	T3	0,0696	0,0696	0,0696	0,0696	0,0696139	0,0697
A	A1	0,4587	0,4587	0,459	0,4587	0,4587288	0,458
	A2	0,5413	0,5413	0,541	0,5413	0,5412712	0,542
G	F	0,2363	0,2363	0,236	0,2363	0,2363494	0,2363
	M	0,7637	0,7637	0,764	0,7637	0,7636505	0,7637
M	M1	0,0632	0,0632	0,0632	0,0632	0,0632448	0,0633
	M2	0,0892	0,0892	0,0892	0,0892	0,0892443	0,0892
	M3	0,4817	0,4817	0,482	0,4817	0,4817322	0,4813
	M4	0,3489	0,3489	0,349	0,3490	0,3489580	0,3494
	M5	0,0168	0,0168	0,0168	0,0168	0,0168204	0,0168
C	C1	0,4285	0,4287	0,429	0,4287	0,4286588	0,4309
	C2	0,5715	0,5713	0,571	0,5713	0,5713411	0,5691
P	No	0,8541	0,8533	0,853	0,8533	0,8533116	0,8533
	Yes	0,1459	0,1467	0,147	0,1467	0,1466883	0,1467
R	R1	0	0	0	0	0	0
	R2	1	1	1	1	1	1

Як показують результати моделювання, типовий профіль неблагонадійного клієнта виглядає наступним чином: зайнятий на повний робочий день (87%), молодше 36 років (54%), чоловічої статі (76%), не має поручителя за кредитом (85%), одружений (48%) або самотній (34%).

Кредитоспроможність клієнта банку залежить від сукупного впливу багатьох критеріїв, зокрема віку, сімейного стану, типу зайнятості тощо. У табл. 4.35 проаналізовано випадки, коли клієнтом банку є приватний підприємець ( $T=T3$ ), вік якого більше 36 років ( $A=A1$ ), вдівець ( $M=M5$ ).

Таблиця 4.35

Результати обчислень станів вершин мережі Байєса різними програмами для задачі кредитного скорингу у випадку інстанційовання вершин  $T=T3$ ,  $A=A1$ ,  $M=M5$

Вершина	Стан	Результати обчислень станів вершин програмами					
		LS-метод	BayesNet	Netica	MSBNx	GeNie	Hugin
T	T1	0	0	0	0	0	0
	T2	0	0	0	0	0	0
	T3	1	1	1	1	1	1
A	A1	1	1	1	1	1	1
	A2	0	0	0	0	0	0
G	F	0,8493	0,864	0,864	0,8640	0,8640054	0,864
	M	0,1507	0,136	0,136	0,1360	0,1359945	0,136
M	M1	0	0	0	0	0	0
	M2	0	0	0	0	0	0
	M3	0	0	0	0	0	0
	M4	0	0	0	0	0	0
	M5	1	1	1	1	1	1
C	C1	0,4762	0,4	0,4	0,4	0,4	0,4
	C2	0,5237	0,6	0,6	0,6	0,6	0,6
P	No	0,44	0,44	0,44	0,44	0,44	0,44
	Yes	0,56	0,56	0,56	0,56	0,56	0,56
R	R1	0,9556	0,9546	0,955	0,9546	0,9545675	0,9546
	R2	0,0444	0,0454	0,0454	0,0454	0,0454324	0,0454

На основі результатів обчислень можна зробити висновок, що приватній підприємець у віці старше 36 років, вдівець має схильність до неповернення кредиту у 4,4 %, що менше, аніж загальний рівень неповернення кредиту по всій вибірці (5,3%).

Тобто, цю категорію клієнтів можна вважати найбільш надійною за умови, що для фінансової установи є прийнятним такий рівень ризику.

Якщо ж клієнтом банку є приватний підприємець ( $T=T3$ ), молодший 36 років ( $A=A2$ ), неодружений ( $M=M4$ ), сума кредиту, потрібного йому більше 2100 грн. ( $C=C2$ ), то ситуація щодо оцінки його кредитоспроможності буде іншою (табл. 4.36)

Таблиця 4.36

Результати обчислень станів вершин мережі Байєса різними програмами для задачі кредитного скорингу у випадку інстанційовання вершин  
 $T=T3, A=A2, M=M4, C=C2$

Вершина	Стан	Результати обчислень станів вершин програмами					
		LS-метод	BayesNet	Netica	MSBNx	GeIe	Hugin
T	T1	0	0	0	0	0	0
	T2	0	0	0	0	0	0
	T3	1	1	1	1	1	1
A	A1	0	0	0	0	0	0
	A2	1	1	1	1	1	1
G	F	0,2411	0,2638	0,264	0,2638	0,264	0,2638
	M	0,7589	0,7362	0,736	0,7362	0,736	0,7362
M	M1	0	0	0	0	0	0
	M2	0	0	0	0	0	0
	M3	0	0	0	0	0	0
	M4	1	1	1	1	1	1
	M5	0	0	0	0	0	0
C	C1	0	0	0	0	0	0
	C2	1	1	1	1	1	1
P	No	0,44	0,44	0,44	0,44	0,44	0,44
	Yes	0,56	0,56	0,56	0,56	0,56	0,56
R	R1	0,8945	0,8963	0,896	0,8963	0,896	0,8963
	R2	0,1055	0,1037	0,104	0,1037	0,104	0,1037

Отже, якщо до банку за наданням кредиту звертається приватний підприємець, молодший 36 років, неодружений, якому потрібен кредит у сумі, що перевищує 2100 грн, то ситуація щодо повернення кредиту є кращою.

Якщо ж ( $C=C2$ ), має місце ймовірність неповернення кредиту 10,6%, що майже у двічі перевищує середнє значення банкрутства (5,3%) за всією вибіркою.

Згідно розрахунків, найбільш відповідальними щодо повернення кредитів є найменш соціально захищені клієнти банку. Тобто у випадках, коли клієнтами банку є пенсіонери, віком більше 56 років, жінки, що офіційно не перебувають у шлюбі ( $T=T2, A=A1, M=M1, G=F, P=Yes$ ), ситуація щодо повернення кредитів є найкращою (табл. 4.37).

Таблиця 4.37

Результати обчислень станів вершин мережі Байєса різними програмами для задачі кредитного скорингу у випадку інстанційовання вершин  $T=T2$ ,  $A=A1$ ,  $M=M1$ ,  $G=F$ ,  $P=Yes$

Вершина	Стан	Результати обчислень станів вершин програмами					
		LS-метод	BayesNet	Netica	MSBNx	GeNie	Hugin
T	T1	0	0	0	0	0	0
	T2	1	1	1	1	1	1
	T3	0	0	0	0	0	0
A	A1	1	1	1	1	1	1
	A2	0	0	0	0	0	0
G	F	1	1	1	1	1	1
	M	0	0	0	0	0	0
M	M1	1	1	1	1	1	1
	M2	0	0	0	0	0	0
	M3	0	0	0	0	0	0
	M4	0	0	0	0	0	0
	M5	0	0	0	0	0	0
C	C1	0,699	0,63	0,63	0,63	0,63	0,63
	C2	0,301	0,37	0,37	0,37	0,37	0,37
P	No	0	0	0	0	0	0
	Yes	1	1	1	1	1	1
R	R1	0,99	0,99	0,99	0,99	0,99	0,99
	R2	0,01	0,01	0,01	0,01	0,01	0,01

Отже, як показали розрахунки, пенсіонери у віці більше 56 років, жінки, що перебувають у громадянському шлюбі та мають поручителя, за кредитом не повертають гроші лише в 1% випадків. Слід відзначити, що серед результатів, наведених у таблицях 4.34-4.37, це найкращі показники. Тобто дана категорія клієнтів дійсно може вважатися високонадійною.

#### 4.4. Порівняння архітектури алгоритмів кластеризації

Після докладного розгляду прикладу роботи модифікованого LS-методу та отриманих результатів роботи доцільно порівняти *оптимальність архітектури алгоритмів кластеризації* Lauritzen-Spiegelhalter (LS), Hugin, і Shenoy-Shafer між собою. Для порівняння були використані критерії:

- оптимальності графічної структури для розповсюдження повідомлення;
- схема проходження повідомлення;
- ефективність використання об'єму пам'яті;
- обчислювальна ефективність кожної архітектури.

Порівнюючи графічні структури різних архітектур щодо ефективності розповсюдження повідомлення, слід відзначити їх топологічні особливості. Зокрема в архітектурах LS і Hugin розповсюдження відбувається в об'єднаному дереві, а в архітектурі SS – у двійковому зв'язаному дереві. Вершини об'єданого дерева – це кліки триангульованого моралізованого графу оригінальної мережі Байєса. Відповідне двійкове зв'язане дерево містить у собі ці кліки та їх деякі підмножини. Тому двійкове зв'язане дерево має більше вершин, ніж відповідне об'єдане дерево.

Об'єдане дерево в архітектурі LS містить тільки таблиці для клік, а в архітектурі Hugin – таблиці клік і сепараторів. Оскільки кінцевим завданням є обчислення значень окремих змінних, в обох архітектурах необхідно виконати для цього певні обчислення. У LS-архітектурі краще робити це, починаючи з таблиці для найменшої кліки, яка містить змінну. Проте ідентифікація найменшої кліки також потребує деяких обчислень. В архітектурі Hugin, якщо змінна міститься в сепараторі, то найвигідніше обчислювати її значення з таблиці найменшого сепаратора, який її містить. Якщо ж змінна не міститься в жодному сепараторі, то обчислення виконуються з таблиці найменшої кліки, що містить цю змінну. Визначення того, чи міститься змінна в сепараторі та ідентифікація найменшого сепаратора або найменшої кліки, що містить змінну, також потребує деяких обчислень. Якщо при побудові двійкового зв'язного дерева в архітектурі SS у нього було включено усі окремі змінні як підмножини, то по завершенню розповсюдження повідомлення можна обрахувати їх значення без подальших обчислень.

Загалом для архітектур LS і Hugin необов'язково використовувати саме об'єдані дерева. Можна скористатись будь-яким зв'язним деревом, у тому числі, двійковими зв'язними деревами. Проте отримані схеми передачі повідомлень можуть бути неефективними відносно як обчислення, так і зберігання на зв'язних деревах з багатьма вершинами. Також необов'язково використовувати двійкове зв'язне дерево для архітектур SS. Для неї можна використовувати будь-яке об'єдане дерево, але це також неефективно з точки зору продуктивності.

Тому при побудові мереж Байєса з LS-архітектурою переважно використовують об'єдані дерева, архітектурою Hugin – об'єдані дерева з сепараторами, а з архітектурою SS – двійкові зв'язні дерева.

Досліджуючи *схеми розповсюдження повідомлень*, перш за все звертають увагу на напрямки розповсюдження повідомлень та топологію відповідної мережі.

В архітектурі LS спочатку довільно визначається кліка об'єданого дерева як корінь. Розповсюдження повідомлень відбувається в два етапи – сходження догори, де кожна кліка посилає повідомлення до своїх батьків, і сходження донизу, де кожна кліка відправляє повідомлення своїм нащадкам. Кожна кліка в об'єданому дереві містить таблицю умовних ймовірностей. Обчислення ймовірностей окремих змінних виконується за допомогою клік.

В архітектурі Hugin аналогічно довільно визначають вершину як корінь. Кожна кліка відправляє повідомлення кожному із сепараторів між нею та її сусідами. Коли сепаратор отримує повідомлення від однієї із своїх сусідніх клік, він відправляє це повідомлення своїй іншій сусідній кліці. Коли усі повідомлення надіслані, в кожній кліці і в кожному сепараторі зберігається таблиця умовних ймовірностей. Обчислення ймовірностей окремих змінних виконується за допомогою клік та сепараторів.

В архітектурі SS вершини, значення яких необхідно обчислити, надсилають

запити на повідомлення усім своїм сусідам. Коли вершина отримує запит на повідомлення, вона, в свою чергу, надсилає запит на повідомлення усім своїм сусідам. Коли усі замовлені повідомлення доставлено, значення бажаних вершин обчислені. Кожна вершина може не зберігати ніякої таблиці, зберігати одну (вхідну або вихідну) або дві та більше (вхідну або вхідні і вихідну). Кожен сепаратор між двома вершинами може зберігати одну або дві таблиці. Обчислення ймовірностей окремих змінних робиться виключно за допомогою вершин.

**Ефективність використання об'єму пам'яті** також різна для різних архітектур. В архітектурі LS кожна кліка в об'єднаному дереві зберігає одну таблицю. Таким чином, загальна потреба обсягів пам'яті залежить від кількості клік в об'єднаному дереві і розмірності станів клік. Якщо після розповсюдження повідомлень в об'єднаних деревах буде отримано нове спостереження, то необхідно починати розповсюдження знову із вхідної таблиці та таблиці спостережень. Не виняток, що виникне потреба редагувати ці таблиці. За цих двох причин необхідно також включати їх у потреби для пам'яті. Також після розповсюдження повідомлення буде одержано таблицю умовних ймовірностей тільки для клік, але оскільки метою є обчислення значень для кожної змінної, їх визначають за допомогою таблиці кліки, тому необхідно передбачати додаткові обсяги пам'яті, що буде виділена для таблиці умовних ймовірностей кожної змінної.

В архітектурі Hugin кожна кліка в об'єднаному дереві і кожний сепаратор між двома сусідніми кліками зберігають одну таблицю. Можливо, що користувачеві знадобиться відредагувати вхідну таблицю та таблицю для спостереження, і вони мають зберігатися окремо. Тому необхідно також передбачати потреби у додатковій пам'яті для зберігання цих таблиць. Після розповсюдження повідомлень, будуть одержані таблиці тільки для клік та сепараторів, тому знадобиться додатково зарезервувати пам'ять для таблиць кожної змінної.

В архітектурі SS кожна вершина може не зберігати ніякої таблиці, зберігати одну (вхідну або вихідну) або дві і більше (вхідну або вхідні і вихідну). Якщо вершина має щонайменше одну вхідну таблицю і необхідно обрахувати її значення, то така вершина повинна мати дві або більше таблиць, а якщо вершина не має вхідної таблиці і значення її не потрібні, то і таблиці в ній також не потрібні.

У всіх інших випадках вершина має одну таблицю (або вхідну, або вихідну). Кожен сепаратор між двома сусідніми вершинами має одну або дві таблиці, залежно від того, які повідомлення знадобляться. Якщо обидві сусідні вершини запитують повідомлення одна від одної, то сепаратор має дві таблиці, а якщо повідомлення потрібне тільки одному з сусідів, то відповідно - одну.

Для визначення обсягів пам'яті, необхідної для реалізації різних архітектур, виконано розрахунки показників, спільних для всіх архітектур та специфічні, індивідуальні показники. Для розрахунків обсяги пам'яті, необхідні для реалізації конкретної архітектури, застосовують змінні типу числа з плаваючою точкою або комою (fpr). Тобто для зберігання таблиці кліки, що складається з трьох бінарних змінних, необхідно  $2^3=8$  fpr.

Для всіх трьох досліджуваних архітектур необхідно мати однаковий об'єм пам'яті для:

- вхідних таблиць  $3+2+2+5\cdot 2^2+3\cdot 2^2+2+2^4=57$  (fpr);
- спостереження:  $2+2=4$  (fpr);
- вихідних таблиць:  $3+2+2+2+5+2+2=18$  (fpr).

Для архітектури LS для зберігання чотирьох клік необхідно додатково  $2^4+2^3+3 \cdot 2^2+5 \cdot 2^2=56$  (fpn).

Для архітектури Hugin слід перебачити  $56+3 \cdot 2^2=68$  (fpn) пам'яті на сепаратори.

Для архітектури SS загальний об'єм пам'яті необхідний для зберігання в сепараторах:

$$2 \cdot (3+2+2+2+3 \cdot 2+2^2+2 \cdot 5+2 \cdot 5+2+5+2+2+2+2^2+2^2+2^2+2^2+2^3+2^3+2^4+2)=204 \text{ (fpn)}.$$

Наприклад, досліджуючи ефективність використання пам'яті для різних архітектур у випадку  $P = No$  (немає поручителя) і  $G = M$  (стать – чоловіча) задачі кредитного скорингу, в результаті виконання аналогічних розрахунків було одержано досить різні значення даного показника (табл. 4.38).

Таблиця 4.38

Порівняння ефективності використання об'єму пам'яті для трьох архітектур

Архітектура	LS	Hugin	SS
Результати для задачі кредитного скорингу у разі інстанціювання вершин $P=No$ і $G=M$ , fpn	135	147	283

Загалом архітектура Hugin завжди потребує більшого об'єму пам'яті, ніж LS у зв'язку з необхідністю зберігати таблиці сепараторів. Порівнюючи архітектури Hugin та SS, не можна дійти однозначних висновків. Хоча двійкове зв'язне дерево має більше вершин, ніж відповідне об'єднане дерево, але не кожна вершина у двійковому зв'язному дереві має таблицю. Вхідні, вихідні таблиці та таблиці спостережень потребують однакового об'єму пам'яті для реалізації обох архітектур, а різниця з'являється лише щодо обсягів, необхідних для зберігання таблиць в кліках та сепараторах для архітектури Hugin і тільки в сепараторах – для архітектури SS. В архітектурі Hugin всі сепаратори мають по одній таблиці. А в архітектурі SS більшість сепараторів мають по дві таблиці і зазвичай кількість сепараторів в двійковому зв'язному дереві набагато більша, ніж у відповідному об'єднаному, однак кожна кліка в об'єднаному дереві зберігає таблицю, а в архітектурі SS їх немає.

З таблиці 4.38 видно, що для прикладу обчислення кредитоспроможності особи, обробка SS-архітектури потребує набагато більших об'ємів пам'яті, ніж LS та Hugin. Однак можна навести приклад, де результати будуть протилежними.

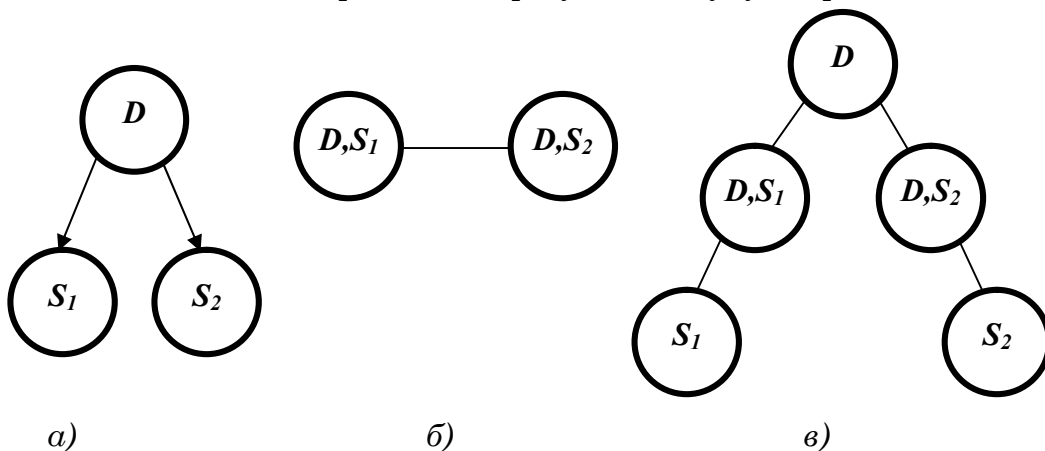


Рис.4.15. Мережа Байєса: (а), об'єднане дерево (б), двійкове зв'язне дерево(в)

Розглянемо мережу Байєса з трьох змінних, зображену на рис. 4.15, де  $D$  – хвороба;  $S_1$  та  $S_2$  – її симптоми. Нехай, кожна з трьох змінних може перебувати в

одному з п'яти станів. Тоді для всіх трьох архітектур потреба в обсягах пам'яті, необхідних для зберігання вхідних таблиць буде:  $5+5^2+5^2=55$ , для зберігання спостережень для двох симптомів:  $5+5=10$ , а для вихідних таблиць –  $3\cdot5=15$  (fpn).

Для архітектури LS з двох клік необхідно додатково ще  $2\cdot25=50$ , в архітектурі Hugin – для двох клік і одного сепаратора  $50+5=55$  (fpn), а в SS –  $4\cdot2\cdot5=40$  (fpn) для чотирьох сепараторів. Таким чином, з усіх досліджуваних архітектур, архітектура SS потребує меншого об'єму пам'яті, ніж Hugin та LS.

Обчислювальна складність трьох розглянутих архітектур є експоненційною за розміром (кількістю змінних) найбільшої кліки. Для архітектури LS обчислювальна складність алгоритму є  $O(n, r^w)$ , Hugin –  $O(N \cdot \exp(w))$ , а SS –  $O(\alpha \cdot \exp(w))$ , де  $\alpha = \sum_i n_i^2$ ,  $n$  – кількість вершин;  $k$  – максимальна кількість батьків вершини;  $r$  – максимальна кількість станів вершини;  $w$  – максимальна кількість вершин у кліці мінус 1;  $N$  – кількість клік;  $n_i$  – кількість сусідів  $i$ -ї кліки [143].

Порівняємо їх за кількістю бінарних арифметичних операцій додавання, множення і ділення, не розглядаючи операцій звернення до таблиць, порівняння, читання/запису та інших. Архітектура Hugin завжди виконує менше операцій додавання, ніж LS тому, що в LS обчислення значень окремих змінних завжди виконується за допомогою таблиць клік. Тоді як в Hugin для деяких змінних використовуються сепаратори, а для деяких – кліки. Тут доводиться нехтувати ціною знаходження найменшої кліки в LS та найменших кліки і сепаратора в Hugin.

В архітектурах LS та Hugin завжди виконується однакова кількість операцій множення, але Hugin завжди відрізняється меншою кількістю ділень. В LS операції ділення відбуваються в кліках, а в Hugin – у сепараторах.

Архітектура SS, на відміну від LS та Hugin, не використовує операцій ділення. Якщо порівнювати архітектури Hugin та SS, то у деяких випадках за архітектури SS виконується менше додавань, а в інших – навпаки. Кількість множень в Hugin та SS не можна порівнювати без врахування операцій ділення в Hugin, тому що вони виконуються для того, щоб уникнути деяких операцій множення. Якщо порівнювати сумарну кількість операцій множення та ділення з коригуючим коефіцієнтом у Hugin та кількістю множень у SS, то у деяких випадках кращою є Hugin, а в деяких – SS.

Для розглянутого прикладу оцінювання кредитоспроможності фізичних осіб, кількість операцій та витрати часу на обробку даних для порівнюваних архітектур представлені у табл. 4.39.

Таблиця 4.39

Кількість операцій і часові затрати

Метод	Кількість операцій, од.	Час, мс
LS	108	35
Hugin	80	26
SS	120	39

Отже, порівнюючи архітектури LS і Hugin, можна зробити висновок, що архітектура Hugin ефективніша за обчисленнями, тоді як LS – за необхідними об'ємами пам'яті. Щодо порівняння ефективності цих архітектур з SS, не можна



зробити ніяких загальних тверджень ні стосовно необхідного загального обсягу пам'яті, ні стосовно обчислювальної ефективності, оскільки у деяких задачах кращі показники має одна архітектура, а у деяких – інша.

### Контрольні питання

1. Назвіть особливості архітектури LS.
2. Опишіть правила "сходження догори".
3. Що таке моралізація графа? Наведіть приклад.
4. Дайте означення об'єданого дерева? Наведіть приклад.
5. Що таке триангуляція графа? Наведіть приклад.
6. Що таке дерево клік (дерево суміжності)? Наведіть приклад.
7. Поясніть алгоритм пропагації в LS-методі – догори та донизу. Наведіть приклади.
8. Опишіть послідовність розрахунку ймовірностей станів вершин.
9. Які графічні структури використовують для розповсюдження повідомлень в архітектурах LS і Hugin?
10. Назвіть, чим вирізняється архітектура Shenoy-Shafer.
11. Яка ефективність використання об'єму пам'яті LS-методу при реалізації на комп'ютері?
12. Опишіть етапи виконання алгоритму розповсюдження.
13. Назвіть послідовність дій під час заповнення об'єданого дерева таблицями.
14. Поясніть формати чисел з фіксованою та плаваючою комами.
15. Яка обчислювальна ефективність LS-методу?
16. Перелічіть всі етапи LS-методу.
17. Назвіть модифікації LS-методу.
18. Наведіть приклади практичного використання LS-методу.
19. Наведіть приклади використанні різних програмних продуктів для дослідження результатів роботи LS-методу.
20. Опишіть критерії порівняння архітектур алгоритмів кластеризації.

## РОЗДІЛ 5 ПОБУДОВА СТРУКТУРИ БАЙЄСІВСЬКОЇ МЕРЕЖІ

- Методи побудови структури байєсівських мереж
- Розрахункова складність задачі побудови байєсівської мережі та методи її зменшення
- Функція Купера-Герсковича
- Модифікована логарифмічна функція Купера-Герсковича
- Функція опису мінімальною довжиною
- Порівняння ефективності оціночних функцій
- Алгоритм евристичного методу побудови байєсівських мереж за навчальними даними
- Методи оцінювання якості побудови байєсівських мереж

### 5.1. Методи побудови структури байєсівських мереж

Існує багато методів розв'язання задачі вибору оптимальної структури мережі Байєса. Оптимальною у тому сенсі, що обрана структура максимально правдоподібно відповідає процесу, що моделюється, або навчальним даним. Взагалі термін "побудова мережі Байєса" в англійській літературі у широкому сенсі означає реалізацію таких процесів:

- пошук оптимальної структури мережі Байєса, тобто спрямованого ациклічного графа, що найбільш адекватно відповідає навчальним даним або досліджуваному процесу;
- обчислення значень таблиць умовних ймовірностей мережі Байєса для вузлів цього графа.

Більшість існуючих методів побудови структури мережі Байєса можна умовно розділити на дві категорії [37, 38]:

- на основі оціночних функцій (search & scoring);
- застосування тестів на умовну незалежність (dependency analysis);
- інші методи, що не ввійшли до даних груп.

Під час вибору способу побудови структури мережі Байєса більшістю із існуючих методів мають місце наступні проблеми.

1. Наявність упорядкованої множини вершин. У більшості методів, особливо відносно старих, вважається, що впорядкована множина вершин задана, але на практиці при роботі з реальними даними це дуже часто не відповідає дійсності.

2. Низька обчислювальна ефективність. Деякі сучасні методи працюють без використання впорядкованої множини вершин, замість неї використовується тест на умовну незалежність. Однак у цьому випадку необхідно виконати експоненціальну кількість таких тестів, що призводить до зменшення ефективності роботи методу у зв'язку із зростанням обсягу обчислень.

3. Проблема побудови великих мереж Байєса. Існує декілька методів здатних побудувати структуру мережі Байєса з декількома сотнями вершин, використовуючи навчальну вибірку з мільйонів записів (наприклад, методи Tetrad II [144] та SopLeq [145]).

*Методи побудови структури мережі Байєса на основі оціночних функцій (search & scoring) розроблені достатньо давно. Чу і Ліу (Chow and Liu) ще в 1968 році*

запропонували алгоритм для побудови мережі Байєса у вигляді дерева [146], заснований на використанні значень взаємної інформації між вершинами. В якості рішення метод видає структуру із значенням спільного розподілу ймовірностей мережі, що найбільше відповідає навчальним даним. Побудова структури мережі Байєса здійснюється за  $O(N^2)$  кроків, де  $N$  – кількість вершин мережі; однак цей алгоритм не працює для багатозв'язаних мереж Байєса.

У 1988 році Рібан і Перл (Rebane and Pearl) запропонували удосконалити змінений алгоритм Чу і Ліу для побудови мережі Байєса у вигляді полі-дерева [147]. Купер і Герскович (Cooper and Herskovits) в 1990 році розробили алгоритм Кутато (Kutato) [148]. На етапі ініціалізації алгоритму вважається, що всі вершини мережі Байєса незалежні, після чого обчислюється ентропія цієї мережі. Потім виконується додавання дуг між вершинами в мережі таким чином, щоб мінімізувати її ентропію. Для роботи алгоритму потрібна наявність впорядкованої множини вершин.

Купер і Герскович [149] у 1992 році запропонували широко відомий алгоритм К2, який виконує пошук структури з максимальним значенням функції Купера-Герсковича (КГ). Для роботи алгоритму потрібно також мати упорядковану множину вершин.

У 1994 році було запропоновано алгоритм HGS [150]. Цей алгоритм суттєво відрізняється від інших алгоритмів тим, що при застосуванні оціночних функції саме в ньому вперше було використано поняття параметричної модульності (parametric modularity) та рівнозначності подій (event equivalence). Інші дослідники тривалий час одночасно не використовували цих понять. Але таке їх застосування дає можливість поєднувати використання статистичної інформації та експертних знань для побудови мережі Байєса.

Вонг і Ксіанг (Wong and Xiang) [151] запропонували у 1994 році алгоритм побудови марковських мереж з використанням значення ентропії та  $I$ -мар. Свою назву даний алгоритм одержав від словосполучення independency map (скорочено  $I$ -map) – незалежна карта. Граф  $G$  ймовірнісної моделі  $M$  називають незалежною картою, якщо з незалежності вершин графа  $G$  випливає незалежність моделі  $M$ . Цей алгоритм дозволяє представити процес, який моделюється, у вигляді  $I$ -мар і у випадку, коли мережа є однозв'язною, мережа Байєса може бути гарантовано побудованою. Разом із Чу (Chu) Ксіанг розробив у 1997 році більш швидкодійний варіант запропонованого алгоритму [152].

Алгоритм Лема-Бахуса (Lam-Bacchus) [153], запропонований у 1996 році, будує евристичну структуру мережі, використовуючи значення взаємної інформації між вершинами, а в якості функції оцінювання використовується функція опису мінімальною довжиною (minimum description length).

Алгоритм Бенедикта (Benedict) [154], запропонований у 1996 році, виконує евристичний пошук на основі впорядкованої множини вершин, аналізуючи умовні незалежності в структурі мережі на основі  $d$ -розподілення, а в якості функції оцінювання використовується ентропія.

СВ-алгоритм [155] запропоновано в 1995 році. Він використовує для побудови впорядкованої множини вершин таблиці умовних ймовірностей між вершинами мережі. Для побудови структури мережі використовується функція Купера-Герсковича.

Алгоритм Фрідмана-Голдшмідта (Friedman-Goldszmidt) [156] запропонований

у 1996 році. Для побудови мережі використовується аналіз її локальних підструктур, а в якості функції оцінювання – функція опису мінімальною довжиною та оцінка Байєса.

В алгоритмі WKD [157], запропонованому в 1996 році, в якості функції оцінювання при побудові мережі використовується функція повідомлення мінімальної довжини (minimum message length), що схожа на функцію опису мінімальною довжиною.

В основу алгоритму Сузукі (Suzuki) [139, 140], запропонованого в 1999 році, для задання послідовності побудови структури мережі покладено метод гілок та границь (branch and bound method), а в якості функції оцінювання використовується опис мінімальної довжини.

Також існує ціла низка різноманітних поглинаючих алгоритмів (greedy algorithm), у яких для оцінювання якості побудови структури мережі можуть використовуватися різноманітні функції, наприклад, максимальної правдоподібності або байєсівський інформаційний критерій [158].

### ***Методи побудови структури мережі з використанням тестів на умовну незалежність (dependency analysis)***

У 1983 році Н. Вермут і С. Лауритцен [159] (Wermuth and Lauritzen) запропонували алгоритм для побудови структури мережі Байєса із використанням таблиць умовних ймовірностей. Цей алгоритм виконує послідовний перебір впорядкованої множини вершин. Для кожної пари вершин  $X_k$  та  $X_t$ , таких, що  $X_t < X_k$  (тобто  $X_k$  це предок відносно  $X_t$ ), виконується обчислення значення умовної незалежності. Цей алгоритм гарантує побудову мережі Байєса за навчальними даними, але при цьому потрібно будувати значну кількість тестів на умовну залежність між вершинами, що з практичної точки зору можливо лише у випадку, коли мережа складається з невеликої кількості вершин.

У 1988 році Перл (Pearl) запропонував алгоритм побудови скінченного спрямованого ациклічного графа (boundary DAG algorithm) [24]. Цей алгоритм буде мережу Байєса за упорядкованою множиною вершин та функцією спільного розподілу (або достатньо великою навчальною вибіркою даних). Разом із будь-яким, не досить складним методом пошуку, цей алгоритм позбавлений проблеми, яка полягає в необхідності виконання великої кількості тестів на умовну залежність із застосуванням алгоритму Вермута і Лоуренса [159]. Однак дана проблема з'являється при застосуванні цього алгоритму для побудови марковських мереж, тобто мереж із прихованими вузлами (hidden nodes).

У 1990 році запропоновано алгоритм SRA [160], який є модифікацією алгоритму скінченного спрямованого ациклічного графа [24]. Цей алгоритм має менш жорсткі вимоги до упорядкованої множини вершин: для побудови мережі Байєса достатньо мати частково упорядковану множину вершин та ще деякі обмеження. Побудова мережі Байєса виконується послідовним додаванням дуг між вершинами, використовуючи для цього евристичний пошук. Недоліком алгоритму є експоненційна кількість розрахунків тестів на умовну ймовірність.

Алгоритм “Конструктор” (constructor algorithm) [161], запропонований в 1990 році дуже схожий на алгоритм побудови скінченного спрямованого ациклічного графа [24], але замість мережі Байєса він намагається побудувати марковську мережу.

Відмінність цього методу від інших, що використовують тести на умовну незалежність, полягає в тому, що він не виконує надлишкові тести на умовну незалежність і йому не потрібна впорядкована множина вершин.

Алгоритму SGS [162], запропонованому у 1990 році, для побудови структури не потрібна наявність впорядкованої множини вершин, але замість неї йому доводиться виконувати кількість тестів на умовну незалежність між вершинами, що зростає за експоненційним законом.

РС-алгоритм, розроблений в 1991 році [163], є удосконаленим варіантом алгоритму SGS. Цей алгоритм розроблявся спеціально для побудови розріджених мереж Байєса (sparse BN), тобто для мереж з невеликою кількістю дуг між вершинами.

Алгоритм KDB, запропонований в 1996 році для визначення напряму побудови мережі, використовує значення взаємної інформації. В якості оціночної функції використовується функціонал, що мінімізує значення мережі.

Алгоритм FBC (full Bayesian network), запропонований в 2006 році, є модифікацією алгоритму KDB, особливістю якої є те, що в якості функції оцінювання при побудові мережі використовується функція сумарних значень взаємної інформації вершин.

### *Інші методи*

Побудована структура мережі Байєса не завжди з високою адекватністю відповідає процесу, що моделюється. Іноколи це пов'язано з неповнотою даних спостережень або недостатньою визначеністю предметної області. Замість побудови однієї, найкращої структури мережі Байєса деякі алгоритми [162, 164] в якості результату видають кілька мережевих структур.

Іноді дослідник може не мати всієї необхідної інформації про процес, що моделюється. Тобто деякі змінні, що описують фактори, які впливають на процес, відсутні. Такі змінні ще називають прихованими (hidden variables) або латентними змінними (latent variables). Існують алгоритми евристичного пошуку [165, 166], які намагаються враховувати такі приховані змінні при моделюванні.

Для випадку, коли навчальні дані неповні або частина з них неправильна (missing data), були запропоновані алгоритми стиснення границь (bound and collapse) та група алгоритмів, які використовують значення максимального математичного очікування (expectation maximization або скорочено EM).

Метод стиснення границь [167] моделює відсутність даних, припускаючи, що ймовірність відсутніх даних набуває значення в інтервалі від 0 до 1. Тобто виконується аналіз цього інтервалу на відсутність даних за наявною інформацією. Після цього робиться стиснення границь інтервалу до точки за допомогою опуклої комбінації з точок екстремумів, використовуючи інформацію про неповні дані.

Алгоритм максимізації математичного очікування [44] дозволяє знайти локальні оптимальні оцінки максимальної правдоподібності параметрів. Головна ідея алгоритму така: якби були відомі значення всіх вузлів, то навчання (на кроці  $M$ ) було б простим, оскільки була б наявна вся необхідна інформація. Тому на кроці  $E$  робиться обчислення значення очікуваної правдоподібності (expectation of the likelihood), включаючи латентні змінні так, ніби вони отримані в результаті спостережень. На кроці  $M$  робиться обчислення значення максимальної правдоподібності (maximum likelihood estimates) параметрів, використовуючи

максимізацію значень очікуваної правдоподібності, отриманих на кроці *E*. Далі алгоритм знову виконує крок *E* з використанням параметрів, отриманих на кроці *M*, і так далі.

На основі алгоритму максимізації математичного очікування розроблено цілу серію подібних алгоритмів [52, 168]. Так наприклад, структурний алгоритм максимізації математичного очікування (structural EM-algorithm) поєднує в собі стандартний алгоритм максимізації математичного очікування, що оптимізує параметри та алгоритм структурного пошуку моделі відбору. Цей алгоритм буде мережі, ґрунтуючись на штрафних ймовірнісних значеннях, які включають значення, отримані за допомогою байєсівського інформаційного критерію, принципу мінімальної довжини опису, а також значеннях інших критеріїв.

**5.2. Розрахункова складність задачі побудови байєсівської мережі та методи її зменшення**

Побудову мережі Байєса можна виконати "у лоб", тобто простим перебором (exhaustive search) множини всіх можливих нециклічних моделей, з яких обирається модель, найбільш адекватна навчальним даним. Але тоді ця задача набуває поліноміальної складності, тому що при повному переборі кількість всіх моделей дорівнює  $3^{\frac{n(n-1)}{2}} - k_{cycle}$ , де *n* – кількість вершин; *k<sub>cycle</sub>* – кількість моделей з циклами. Кількість всіх можливих нециклічних моделей можна порахувати за допомогою рекурентної формули Робінсона [94, 95]:

$$f(n) = \sum_{i=1}^n (-1)^{i+1} \cdot C_n^i \cdot 2^{i(n-i)} \cdot f(n-i), \tag{5.1}$$

де *n* – кількість вершин, а *f*(0) = 1.

Таблиця 5.1

Таблиця залежності кількості моделей без циклів від кількості вершин, які потрібно проаналізувати при повному переборі моделей

Кількість вершин	Кількість моделей без циклів	Кількість вершин	Кількість моделей без циклів
1	1	8	783.702.329.343
2	3	9	1.213.442.454.842.881
3	25	10	4.175.098.976.430.598.100
4	543	...	
5	29.281	15	2,38*10^41
6	3.781.503	...	...
7	1.138.779.265	20	2,34*10^72

Однак на практиці виконати повний перебір моделей можна тільки для мереж, що мають не більше, ніж сім вершин. При кількості вершин більше семи

виконати простий перебір неможливо, оскільки ця операція потребує значних обчислювальних ресурсів, які фізично забезпечити неможливо.

Тому виникає проблема зменшення розрахункової складності при побудові мереж Байєса.

Для її вирішення використовують методи:

- з використанням упорядкованої множини вершин;
- інші евристичні методи.

### **Методи з використанням упорядкованої множини вершин**

Купер і Герскович [149], Дехтер [95] та багато інших дослідників [24, 148, 155, 159, 160] для зменшення простору структур можливих мереж запропонували вважати, що множина змінних (вершин), описуючих процес, упорядкована.

Тобто задається впорядкована множина вершин вигляду  $\{X^{(1)} < X^{(2)} < \dots < X^{(n)}\}$ , де вершина  $X^{(1)}$  – це головна коренева вершина, у якій немає предків, а запис  $X^{(i)} < X^{(j)}$  означає, що вершина  $X^{(i)}$  передуює вершині  $X^{(j)}$ , тобто вершина  $X^{(j)}$  не може бути предком вершини  $X^{(i)}$ , а дуги можуть спрямовуватись тільки з  $X^{(i)}$  в  $X^{(j)}$ .

Використовуючи попередню упорядкованість вершин, кількість структур, необхідних для розгляду, зменшують до  $2^{\frac{n(n-1)}{2}}$ , де  $n$  – кількість вершин. Це дуже важливе припущення, що дозволяє заощаджувати обчислювальні потужності, при  $n=10$  кількість структур зменшується в  $10^7$  разів. Але при цьому потрібно застосовувати експертів, які повинні докладно знати та розуміти предметну область і будуть виконувати попередню упорядкованість вершин. Також можна упорядковувати вершини на основі аналізу характеристик навчальних даних. Наприклад, можна використати кореляцію атрибутів вершин [169] або пошук типу табу (taboo search) [170]. Але навіть при введенні попередньої упорядкованості вершин, обсяг обчислень залишається величезним, наприклад при  $n=8$  потрібно виконати аналіз  $2^{\frac{8 \cdot (8-1)}{2}} = 268.435.456$  структур.

Метод, відомий у літературі як К2 [149], полягає у виконанні наступної послідовності кроків. До вершини  $X^{(N)}$  по черзі, послідовно перебираючи впорядковану множину вершин, додають предків від  $X^{(1)}$  до  $X^{(N-1)}$  та обчислюють значення функції Купера-Герсковича (КГ) для кожної побудованої таким чином мережі. В якості батьківської для дочірньої вершини  $X^{(N)}$  залишають вершину  $X^{(i)}$ , за якої функція Купера-Герсковича набуває максимального значення. Після цього до вершин  $X^{(N)}$  і  $X^{(N-1)}$  по черзі додають батьківські від  $X^{(1)}$  до  $X^{(N-2)}$ , обчислюючи значення функції Купера-Герсковича відповідних мереж. У результаті утворюється структура мережі, для якої функція Купера-Герсковича набуває максимального значення.

Припущення про наявність упорядкованої множини вершин суттєво скорочує простір всіх можливих ациклічних структур. Але з'являється нова нетривіальна проблема: як отримати впорядковану множину вершин мережі, маючи множину навчальних даних. Найбільш очевидний спосіб – це залучення експертів. Але це не завжди можливо, часто виникає проблема пошуку кваліфікованих експертів, а

інколи може виникнути потреба моделювання даних у такій предметній області, де досвідчених експертів просто немає.

**Інші евристичні методи**

Під евристичними розуміють методи, які дозволяють виконувати побудову мережі Байєса без здійснення повного перебору простору всіх можливих структур мережі. Серед евристичних методів нерідко зустрічаються методи із використанням упорядкованої множини вершин, але на практиці дуже часто вона відсутня. Тому замість упорядкованої множини вершин застосовують інші підходи.

Зокрема замість упорядкованої множини вершин можна застосовувати інші характеристики навчальних даних, а саме: кореляцію між змінними мережі. Але, як свідчать результати обчислювальних експериментів, методи із використанням кореляції неефективні. Так, деякі алгоритми використовують різноманітні тести на умовну незалежність. В якості тесту на умовну незалежність можна використовувати  $\chi^2$  - критерій Пірсона або  $\chi^2$  - тест відношення правдоподібності [171], але найбільш ефективними виявилися методи із використанням значення взаємної інформації (mutual information).

Таблиця 5.2

Таблиця значень частоти взаємного сполучення змінних  $X^{(1)}$  та  $X^{(2)}$

Змінні	$X^{(1)} = x_1^{(1)}$	$X^{(1)} = x_2^{(1)}$	Сума
$X^{(2)} = x_1^{(2)}$	$a$	$b$	$a+b$
$X^{(2)} = x_2^{(2)}$	$c$	$d$	$c+d$
Сума	$a+c$	$b+d$	$n$

У табл. 5.2  $a, b, c, d$  – це частоти взаємного сполучення двох змінних, а  $n$  – сума частот.

Для оцінки виміру щільності залежності використовують різні статистичні показники зв'язку змінних, зокрема коефіцієнти асоціації, контингенції, взаємної спряженості Пірсона, значення взаємної інформації.

**Статистичні показники зв'язку змінних.** В якості тесту на умовну незалежність для виміру щільності залежності застосовують статистичні показники. Наприклад, коефіцієнти асоціації, контингенції (спряженості, contingency) або взаємної спряженості [172].

**Коефіцієнт асоціації Юла** для змінних  $X^{(1)}$  та  $X^{(2)}$  з таблиці 5.2 обчислюється за формулою:

$$K_{ac} = \frac{a \cdot d - b \cdot c}{a \cdot d + b \cdot c} \tag{5.2}$$

**Коефіцієнт контингенції (спряженості) Бравайса** (по-іншому – показник подібності Чупрова) між змінними  $X^{(1)}$  та  $X^{(2)}$  в табл. 5.2 розраховується за формулою:



$$K_{kon} = \frac{a \cdot d - b \cdot c}{\sqrt{(a+b) \cdot (b+d) \cdot (a+c) \cdot (c+d)}}. \quad (5.3)$$

Результати розрахунків згруповані в таблиці 5.3.

Таблиця 5.3

Таблиця значень частоти взаємного сполучення змінних  $X^{(1)}$  та  $X^{(2)}$

Змінні	$X^{(1)} = x_1^{(1)}$	...	$X^{(1)} = x_n^{(1)}$	Сума
$X^{(2)} = x_1^{(2)}$	$k_{11}$	...	$k_{n1}$	$\sum_{j=1}^n k_{j1}$
...	...	...	...	...
$X^{(2)} = x_m^{(2)}$	$k_{1m}$	...	$k_{nm}$	$\sum_{j=1}^n k_{jm}$
Сума	$\sum_{i=1}^m k_{1i}$	...	$\sum_{i=1}^m k_{ni}$	$\sum_{i=1}^m \sum_{j=1}^n k_{ji}$

Коефіцієнти асоціації та контингенції приймають значення від мінус одиниці до одиниці, треба зауважити, що для одних і тих самих даних коефіцієнт контингенції завжди менший коефіцієнта асоціації. Одиниця означає повний позитивний зв'язок, від'ємна одиниця означає повний негативний зв'язок, а нуль означає незалежність змінних.

У випадку, коли змінні мережі мають більше, ніж два стани, як наприклад, змінні  $X^{(1)}$  та  $X^{(2)}$  з табл. 5.3, застосовують коефіцієнт взаємної спряженості.

**Коефіцієнт взаємної спряженості Пірсона** розраховується за формулою

$K = \sqrt{\frac{\varphi^2}{1 + \varphi^2}}$ , де  $\varphi^2$  – показник середньо квадратичної спряженості, який розраховується:

$$\varphi^2 = \left( \sum_{j=1}^n \left( \sum_{i=1}^m \left( \frac{k_{ji}}{\left( \sum_{j=1}^n k_{ji} \right) \cdot \left( \sum_{i=1}^m k_{ji} \right)} \right) \right) \right). \quad (5.4)$$

Коефіцієнт взаємної спряженості Пірсона приймає значення від нуля до одиниці. Нуль вказує на те, що змінні незалежні, а одиниця означає, що значення однієї змінної можна точно спрогнозувати по іншій.

**Значення взаємної інформації.** Для оцінювання ступеня залежності двох довільних змінних  $x^i$  та  $x^j$  при побудові мережі Байеса, Чоу та Ліу [146] запропонували використання значення взаємної інформації  $MI(x^i, x^j)$ .

Для його розрахунку використовується формула 5.5:

$$MI(x^i, x^j) = \sum_{x^i, x^j} P(x^i, x^j) \cdot \log \left( \frac{P(x^i, x^j)}{P(x^i) \cdot P(x^j)} \right). \quad (5.5)$$

За своєю суттю значення взаємної інформації – це аналог кореляції, але за своїм змістом – це оцінка кількості інформації, яка наявна у змінній  $x^i$  про змінну  $x^j$ . Значення взаємної інформації є невід’ємні значення, тобто  $MI(x^i, x^j) = 0$ , а у випадку якщо вершини  $x^i$  та  $x^j$  повністю незалежні одна від одної, то  $MI(x^i, x^j) = 0$ , тому що  $P(x^i, x^j) = P(x^i) \cdot P(x^j)$ , тобто:

$$\log \left( \frac{P(x^i, x^j)}{P(x^i) \cdot P(x^j)} \right) = \log \left( \frac{P(x^i) \cdot P(x^j)}{P(x^i) \cdot P(x^j)} \right) = \log(1) = 0. \quad (5.6)$$

У випадку, коли мережа Байєса складається з  $N$  вершини, для обчислення  $MI(x^i, x^j)$  всіх пар  $x^i$  та  $x^j$  потрібно виконати  $\frac{N \cdot (N-1)}{2}$  обчислень, при цьому потрібно враховувати, що  $MI(x^i, x^j) = MI(x^j, x^i)$ .

Наприклад, необхідно обчислити значення взаємної інформації між змінними  $X^1$  та  $X^2$ , для яких задана множина навчальних даних з 10 записів (табл. 5.4).

Таблиця 5.4

Множина навчальних даних з 10 записів

$n$	$X^1$	$X^2$	$n$	$X^1$	$X^2$
1	0	1	6	0	1
2	1	0	7	1	0
3	0	1	8	1	0
4	1	0	9	0	1
5	0	1	10	1	1

Значення взаємної інформації обчислюється із використанням проміжних значень ймовірностей, наведених у таблицях 5.5, 5.6 та 5.7.

Таблиця 5.5

Таблиця значень ймовірності змінної  $X^1$

$P(X^1)$	
$P(X^1=0)$	$P(X^1=1)$
0,5	0,5

Таблиця 5.6

Таблиця значень ймовірності змінної  $X^2$

$P(X^2)$	
$P(X^2=0)$	$P(X^2=1)$
0,4	0,6

Таблиця 5.7

 Таблиця значень спільної ймовірності змінних  $X^1$  та  $X^2$ 

$P(X^1, X^2)$	
$P(X^1 = 0, X^2 = 0)$	0
$P(X^1 = 0, X^2 = 1)$	0,5
$P(X^1 = 1, X^2 = 0)$	0,4
$P(X^1 = 1, X^2 = 1)$	0,1

$$\begin{aligned}
 MI(X^{(1)}, X^{(2)}) &= \sum_{x^{(1)}, x^{(2)}} P(X^{(i)}, X^{(j)}) \cdot \log \left( \frac{P(X^{(i)}, X^{(j)})}{P(X^{(i)}) \cdot P(X^{(j)})} \right) = \\
 &= P(X^{(1)} = 0, X^{(2)} = 0) \cdot \log \left( \frac{P(X^{(1)} = 0, X^{(2)} = 0)}{P(X^{(1)} = 0) \cdot P(X^{(2)} = 0)} \right) + \\
 &+ P(X^{(1)} = 0, X^{(2)} = 1) \cdot \log \left( \frac{P(X^{(1)} = 0, X^{(2)} = 1)}{P(X^{(1)} = 0) \cdot P(X^{(2)} = 1)} \right) + \\
 &+ P(X^{(1)} = 1, X^{(2)} = 0) \cdot \log \left( \frac{P(X^{(1)} = 1, X^{(2)} = 0)}{P(X^{(1)} = 1) \cdot P(X^{(2)} = 0)} \right) + \\
 &+ P(X^{(1)} = 1, X^{(2)} = 1) \cdot \log \left( \frac{P(X^{(1)} = 1, X^{(2)} = 1)}{P(X^{(1)} = 1) \cdot P(X^{(2)} = 1)} \right) = \\
 &= 0 + 0,5 \cdot \log \left( \frac{0,5}{0,5 \cdot 0,6} \right) + 0,4 \cdot \log \left( \frac{0,4}{0,5 \cdot 0,4} \right) + 0,1 \cdot \log \left( \frac{0,1}{0,5 \cdot 0,6} \right) = \\
 &= 0 + 0,2554 + 0,2773 - 0,1099 = 0,4228 \quad .
 \end{aligned}$$

Ідея використання значення взаємної інформації для побудови мереж Байєса була розвинута пізніше [173, 139, 140]. Так, за евристичним методом [173], спочатку виконують обчислення значення взаємної інформації між всіма вершинами, потім – цілеспрямований пошук, який використовує принцип опису мінімальною довжиною в якості оціночної функції, що застосовується на кожній ітерації алгоритму побудови мережі Байєса.

### 5.3. Функція Купера-Герсковича

Купер і Герскович [149] запропонували метод із використанням однойменної функції для навчання мережі Байєса, який заснований на пошуку структури мережі із максимальним значенням функції Купера-Герсковича. Функція Купера-Герсковича структури  $g \in G$  при заданій послідовності з  $n$  спостережень  $x^n = d_1 d_2 \dots d_n$  записується у вигляді рівняння:

$$P(g, x^n) = P(g) \cdot \prod_{j \in J} \left( \prod_{s \in S(j, g)} \frac{(\alpha^{(j)} - 1)! \cdot \prod_{q \in A^{(j)}} (n[q, s, j, g]!)}{(n[s, j, g] + \alpha^{(j)} - 1)!} \right), \quad (5.7)$$

де  $P(g)$  – апіорна ймовірність структури  $g \in G$ , але взагалі її часто опускають

при обчисленнях, вважаючи, що всі структури рівно ймовірні; запис  $j \in J = \{1, \dots, N\}$  означає перебір всіх вершин структури мережі  $g$ , а  $s \in S(j, g)$  означає перебір множини всіх наборів значень, які набувають батьківські вершини  $j$ -ї вершини:

$$n(s, j, g) = \sum_{i=1}^n I(\pi_i^{(j)} = s); \quad n[q, s, j, g] = \sum_{i=1}^n I(x_i = q, \pi_i^{(j)} = s) \quad (5.8)$$

де  $\pi^{(j)} = \Pi^{(j)}$  означає  $X^{(k)} = x^{(k)}, \forall k \in \phi^{(j)}$ , а функція  $I(E) = 1$ , коли предикат  $E = true$ , інакше  $I(E) = 0$ .

Алгоритм навчання мережі Байєса із використанням функції Купера-Герсковича заснований на циклічному переборі всіх можливих ациклічних мережевих структур.

В  $g^*$  зберігається оптимальна мережна структура, тобто структура, яка максимально відповідає навчальним даним. Оптимальною структурою буде та, що має найбільше значення функції  $P(g, x^n)$ .

**Алгоритм побудови мережі Байєса із використанням метода Купера-Герсковича**

- 1 –  $g^* \leftarrow g_0 (\in G)$ ;
- 2 – для  $\forall g \in G - \{g_0\}$  якщо  $P(g, x^n) > P(g^*, x^n)$  то  $g^* \leftarrow g$ ;
- 3 – в якості рішення використовується  $g^*$ .

В якості прикладу слід навести обчислення значення функції Купера-Герсковича для структури зображеної на рис 5.1.

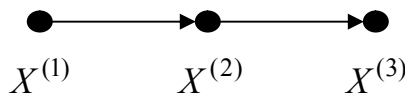


Рис. 5.1. Оптимальна структура мережі Байєса

Навчальні дані наведені у табл.5.8.

Таблиця 5.8

Множина навчальних даних з 10 записів

$n$	$X^{(1)}$	$X^{(2)}$	$X^{(3)}$
1	0	1	1
2	1	0	0
3	0	1	1
4	1	0	0
5	0	1	1
6	0	1	1
7	1	0	1
8	1	0	0
9	0	1	1
10	1	1	1

$$\begin{aligned}
 P(1, g, x^n) &= \frac{(2-1)! \cdot 5! \cdot 5!}{(10+2-1)!} = 0.00036; \\
 P(2, g, x^n) &= \frac{(2-1)! \cdot 0! \cdot 5!}{(5+2-1)!} \cdot \frac{(2-1)! \cdot 4! \cdot 1!}{(5+2-1)!} = 0.0056; \\
 P(3, g, x^n) &= \frac{(2-1)! \cdot 3! \cdot 1!}{(4+2-1)!} \cdot \frac{(2-1)! \cdot 0! \cdot 6!}{(6+2-1)!} = 0.0071; \\
 P(g, x^n) &= \frac{1}{25} \cdot \prod_{j \in J} P(j, g, x^n) = 5.7254 \cdot 10^{-10}.
 \end{aligned}$$

Слід відзначити, що при використанні функції Купера-Герсковича необхідно враховувати обчислювальні обмеження моделюючих систем, пов'язані зі скінченною довжиною розрядної сітки. Наприклад, коли в структурі є дві вершини  $X^{(1)}$  і  $X^{(2)}$ , а множина навчальних даних складається з мільйона записів  $D = \{d^{(1)}, \dots, d^{(1.000.000)}\}$ , то при обчисленні  $P(g, x^n)$  потрібно буде обчислити факторіал вигляду:  $(n[s, j, g] + \alpha^{(j)} - 1)! = (1.000.000 + \alpha^{(j)} - 1)!$ , тоді як такі 32-х розрядні програми MatLab і MathCAD здатні обчислити факторіали не більше 170!.

#### 5.4. Модифікована логарифмічна функція Купера-Герсковича

Для ширшого застосування функції Купера-Герсковича необхідно було зменшити обсяги обчислювальних ресурсів, необхідних для проведення розрахунків: потрібно позбутися факторіала. З цією метою, функцію Купера-Герсковича було модифіковано наступним чином.

Спочатку обчислено логарифм виразу, за допомогою якого визначається функція Купера-Герсковича:

$$\begin{aligned}
 \log(P(g, x^n)) &= \log \left( P(g) \cdot \prod_{j \in J} \left( \prod_{s \in S(j, g)} \frac{(\alpha^{(j)} - 1)! \cdot \prod_{q \in A^{(j)}} (n[q, s, j, g]!)}{(n[s, j, g] + \alpha^{(j)} - 1)!} \right) \right) = \\
 &= \log(P(g)) + \sum_{j \in J} \left( \sum_{s \in S(j, g)} \left( \sum_{i=1}^{\alpha^{(j)}-1} i + \sum_{q \in A^{(j)}} \left( \sum_{i=1}^{n[q, s, j, g]} i \right) - \sum_{i=1}^{n[s, j, g] + \alpha^{(j)} - 1} i \right) \right) = \\
 &= \log(P(g)) + \sum_{j \in J} \left( \sum_{s \in S(j, g)} \left( \sum_{q \in A^{(j)}} \left( \sum_{i=1}^{n[q, s, j, g]} i \right) - \sum_{i=\alpha^{(j)}}^{n[s, j, g] + \alpha^{(j)} - 1} i \right) \right) .
 \end{aligned}$$

Отриманий вираз помножено на мінус одиницю, а для економії обчислювальних ресурсів з нього виключено  $\log(P(g))$ .

Cooper G., Herskovits E. [149] припустили, що апіорні ймовірності  $P(g)$  всіх структур рівні. Тепер замість пошуку структури з максимальним значенням функції Купера-Герсковича потрібно здійснювати пошук структури з мінімальним значенням модифікованої функції Купера-Герсковича (МФКГ):

$$F(g, x^n) = \sum_{j \in J} \left( \sum_{s \in S(j, g)} \left( \sum_{i=\alpha^{(j)}}^{n[s, j, g] + \alpha^{(j)} - 1} i \right) \right) - \sum_{j \in J} \left( \sum_{s \in S(j, g)} \left( \sum_{q \in A^{(j)}} \left( \sum_{i=1}^{n[q, s, j, g]} i \right) \right) \right); \quad (5.9)$$

У ході обчислювальних експериментів виявлено, що функції Купера-Герсковича і модифікована функція Купера-Герсковича видають на одних і тих самих навчальних даних абсолютно однакові рішення. Однак на невеликих (до 10 вершин), функція Купера-Герсковича працює швидше, ніж модифікована функція Купера-Герсковича, а на мережах з більшою кількістю вершин ситуація протилежна.

### 5.5. Функція опису мінімальною довжиною

Згідно з теорією кодування Шеннона, при відомому розподілі  $P(X)$  випадкової величини  $X$ , довжина оптимального коду для передачі конкретного значення  $x$  по каналу зв'язку прямує до  $L(x) = -\log P(x)$ . Ентропія джерела  $S(P) = -\sum_x P(x) \cdot \log P(x)$  є мінімально очікуваною довжиною закодованого повідомлення. Будь-який інший код, заснований на хибному уявленні про джерело повідомлень, призведе до утворення повідомлення більшої довжини. Іншими словами, чим краще побудована модель джерела, тим компактніше кодуються дані.

При навчанні джерелом даних виступає деяка невідома істинна функція розподілу  $P(D|h_0)$ , де  $D = \{d_1, \dots, d_N\}$  – набір даних, а  $h$  – гіпотеза про ймовірнісне походження даних;  $L(D|h) = -\log P(D|h)$  – емпіричний ризик, адитивний за кількістю спостережень і пропорційний емпіричній помилці. Відмінність між  $P(D|h_0)$  і розподілом моделі  $P(D|h)$  за мірою Кулбака-Леблера визначається:

$$\begin{aligned} |P(D|h) - P(D|h_0)| &= \sum_D P(D|h_0) \cdot \log \frac{P(D|h_0)}{P(D|h)} \\ &= \sum_D P(D|h_0) \cdot |L(D|h) - L(D|h_0)| \geq 0, \end{aligned} \quad (5.10)$$

тобто вона представляє собою різницю очікуваної довжини кодування даних на основі гіпотези про мінімально можливу довжину. Ця різниця завжди додатна і дорівнює нулю лише при повному збігові двох розподілів. Іншими словами, гіпотеза краща в тому випадку, коли середня довжина кодування даних є, наскільки це можливо, меншою [174]. Принцип опису мінімальною довжиною у своїй спрощеній і найбільш загальній формі полягає в наступному: серед безлічі моделей варто обрати ту, яка дозволяє описати дані найбільш коротко і без втрати інформації [175].

У загальному випадку задача опису мінімальною довжиною формулюється таким чином. Спочатку задається множина навчальних даних  $D = \{d_1, \dots, d_n\}$ , де  $d_i = \{x_i^{(1)} x_i^{(2)} \dots x_i^{(N)}\}$  (нижній індекс – номер спостереження, а верхній – номер змінної);  $n$  – кількість спостережень, кожне спостереження складається з  $N$  ( $N \geq 2$ ) змінних  $X^{(1)}, X^{(2)}, \dots, X^{(N)}$ ; кожна  $j$ -а змінна ( $j = 1, \dots, N$ ) має  $A^{(j)} = \{0, 1, \dots, \alpha^{(j)} - 1\}$  ( $\alpha^{(j)} \geq 2$ ) станів; кожна структура мережі Байеса представлена  $N$  множинами предків  $(\Pi^{(1)}, \dots, \Pi^{(N)})$ , тобто для кожної вершини  $j = 1, \dots, N$ ,  $\Pi^{(j)}$  – це множина батьківських вершин така, що  $\Pi^{(j)} \subseteq \{X^{(1)}, \dots, X^{(N)}\} \setminus \{X^{(j)}\}$  (вершина не може бути батьком самої себе, тобто петлі у

графі відсутні). Тоді опис мінімальної довжини структури  $g \in G$  при заданій послідовності з  $n$  спостережень  $x^n = d_1 d_2 \dots d_n$  обчислюється за формулою:

$$L(g, x^n) = H(g, x^n) + \frac{k(g)}{2} \cdot \log(n), \quad (5.11)$$

де  $k(g)$  – кількість незалежних умовних ймовірностей у структурі мережі Байєса  $g$ , а  $H(g, x^n)$  – це емпірична ентропія.

$$H(g, x^n) = \sum_{j \in J} H(j, g, x^n), \quad (5.12)$$

$$k(g) = \sum_{j \in J} k(j, g), \quad (5.13)$$

де опис мінімальної довжини  $j$ -ї вершини обчислюється за формулою:

$$L(j, g, x^n) = H(j, g, x^n) + \frac{k(j, g)}{2} \cdot \log(n). \quad (5.14)$$

Тут  $k(j, g)$  – кількість незалежних умовних ймовірностей  $j$ -ї вершини:

$$k(j, g) = (\alpha^{(j)} - 1) \cdot \prod_{k \in \phi(j)} \alpha^k, \quad (5.16)$$

де  $\phi(j) \subseteq \{1, \dots, j-1, j+1, \dots, N\}$ ; це множина, для якої  $\Pi^{(j)} = \{X^{(k)} : k \in \phi(j)\}$ .

Емпірична ентропія  $j$ -ї вершини обчислюється за формулою:

$$H(j, g, x^n) = \sum_{s \in S(j, g)} \sum_{q \in A^{(j)}} -n[q, s, j, g] \cdot \log \frac{n[q, s, j, g]}{n[s, j, g]}, \quad (5.17)$$

де

$$n(s, j, g) = \sum_{i=1}^n I(\pi_i^{(j)} = s), \quad (5.18)$$

$$n[q, s, j, g] = \sum_{i=1}^n I(x_i = q, \pi_i^{(j)} = s), \quad (5.19)$$

де  $\pi^{(j)} = \Pi^{(j)}$ , а  $X^{(k)} = x^{(k)}, \forall k \in \phi^{(j)}$ , функція  $I(E) = 1$ , якщо предикат  $E=true$ , інакше  $I(E)=0$ .

Простий алгоритм навчання мережі Байєса із використанням опису мінімальною довжиною виглядає таким чином: циклічно робиться перебір усіх можливих не циклічних структур мереж; в  $g^*$  зберігається оптимальна структура мережі. Оптимальною структурою буде та, для якої значення функції  $L(g, x^n)$  буде найменшим.

**Алгоритм побудови мережі Байєса із використанням опису мінімальної довжини**

Крок 1:  $g^* \leftarrow g_0 (\in G)$ ;

Крок 2: для  $\forall g \in G - \{g_0\}$  якщо  $L(g, x^n) < L(g^*, x^n)$  то  $g^* \leftarrow g$ ;

Крок 3: в якості рішення приймається  $g^*$ .

**Приклад застосування функції опису мінімальною довжиною**

Нехай задано набір навчальних даних з 10 спостережень для навчання мережі Байєса (табл. 5.8.). У випадку повного перебору всіх можливих структур мережі потрібно буде розглянути 25 структур. Після того, як всі структури будуть розглянуті, в якості оптимальної буде видана структура, зображена на рис. 5.1.

Для обчислення значення опису мінімальною довжиною для цієї структури робляться наступні дії.

У вершини  $X^{(1)}$  відсутні батьки, тобто  $\Pi^{(1)} = \{\}$ . Емпірична ентропія обчислюється так:  $H(j=1, g) = -5 \cdot \log\left(\frac{5}{10}\right) - 5 \cdot \log\left(\frac{5}{10}\right) = 6,9315$ , а кількість незалежних умовних ймовірностей  $k(j=1, g) = 2 - 1 = 1$ . Отже, довжина опису вершини  $X^{(1)}$  дорівнює  $L(1, g) = 6,9315 + \frac{1}{2} \cdot \log(10) = 8,0828$ . При обчисленні можна застосовувати логарифм із будь-якою основою, у даному прикладі використовується з основою  $e=2,7183$ , тобто натуральний логарифм.

Таблиця 5.9

Таблиця значень параметрів вершини  $X^{(1)}$

$X^{(1)}$	$n[q, s, j, g]$	$n[s, j, g]$
0	5	10
1	5	

Вершина  $X^{(2)}$  має одну батьківську вершину  $X^{(1)}$ , тобто  $\Pi^{(2)} = \{X^{(1)}\}$ . Емпірична ентропія обчислюється так:

$$H(j=2, g) = \left(-0 \cdot \log\left(\frac{0}{5}\right) - 5 \cdot \log\left(\frac{5}{5}\right)\right) + \left(-4 \cdot \log\left(\frac{4}{5}\right) - 1 \cdot \log\left(\frac{1}{5}\right)\right) = 2,502,$$

а кількість незалежних умовних ймовірностей  $k(j=2, g) = (2-1) \cdot 2 = 2$ . Отже, довжина опису вершини дорівнює:  $L(2, g) = 2,502 + \frac{2}{2} \cdot \log(10) = 4,8046$ .

Таблиця 5.10

Таблиця значень параметрів вершин  $X^{(2)}$  та  $X^{(3)}$

$X^{(1)}$	$X^{(2)}$	$n[q, s, j, g]$	$n[s, j, g]$	$X^{(2)}$	$X^{(3)}$	$n[q, s, j, g]$	$n[s, j, g]$
0	0	0	5	0	0	3	4
0	1	5		0	1	1	
1	0	4	5	1	0	0	6
1	1	1		1	1	6	

Вершина  $X^{(3)}$  має одну батьківську вершину  $X^{(2)}$ , тобто  $\Pi^{(3)} = \{X^{(2)}\}$ . Емпірична ентропія обчислюється як:

$$H(j=3, g) = \left(-3 \cdot \log\left(\frac{3}{4}\right) - 1 \cdot \log\left(\frac{1}{4}\right)\right) + \left(-0 \cdot \log\left(\frac{0}{6}\right) - 6 \cdot \log\left(\frac{6}{6}\right)\right) = 2.2493,$$



а кількість незалежних умовних ймовірностей  $k(j=3, g) = (2-1) \cdot 2 = 2$ . Отже, довжина опису вершини  $X^{(3)}$  дорівнює:

$$H(j=3, g) = \left( -3 \cdot \log\left(\frac{3}{4}\right) - 1 \cdot \log\left(\frac{1}{4}\right) \right) + \left( -0 \cdot \log\left(\frac{0}{6}\right) - 6 \cdot \log\left(\frac{6}{6}\right) \right) = 2.2493$$

Тобто довжина опису структури представленої на рис. 5.1 дорівнює:

$$H(g, x^n) = \sum_{j=1}^3 H(j, g, x^n) = 17,4393.$$

Докладнішу інформацію про створення та використання опису мінімальною довжиною можна знайти в джерелах [139, 140, 174, 175].

### 5.6. Порівняння ефективності оціночних функцій

Для побудови структури мережі Байєса найчастіше застосовують методи із використанням функцій Купера-Герсковича або опису мінімальною довжиною, а також різні модифікації цих функцій, наприклад, модифікована функція Купера-Герсковича.

Результати обчислювальних експериментів показали, що на коротких навчальних вибірках, до 170 записів, і мережах, що складаються не більш ніж з 10 вершин, функція Купера-Герсковича працює швидше у порівнянні з модифікованою функцією Купера-Герсковича й опису мінімальною довжиною. Але модифікована функція Купера-Герсковича й опису мінімальною довжиною, на відміну від функції Купера-Герсковича, працюють із навчальними вибірками будь-якого розміру. Крім того, в ході досліджень з'ясовано, що методи побудови мереж Байєса із використанням функції Купера-Герсковича та її модифікацій, виконують перенавчання мережі Байєса, тобто такі мережі часто містять зайві дуги. На рис. 5.2 та 5.3 представлені графіки часу, витраченого на побудову мереж Байєса методами, що використовують модифіковану функцію Купера-Герсковича та опису мінімальною довжиною.

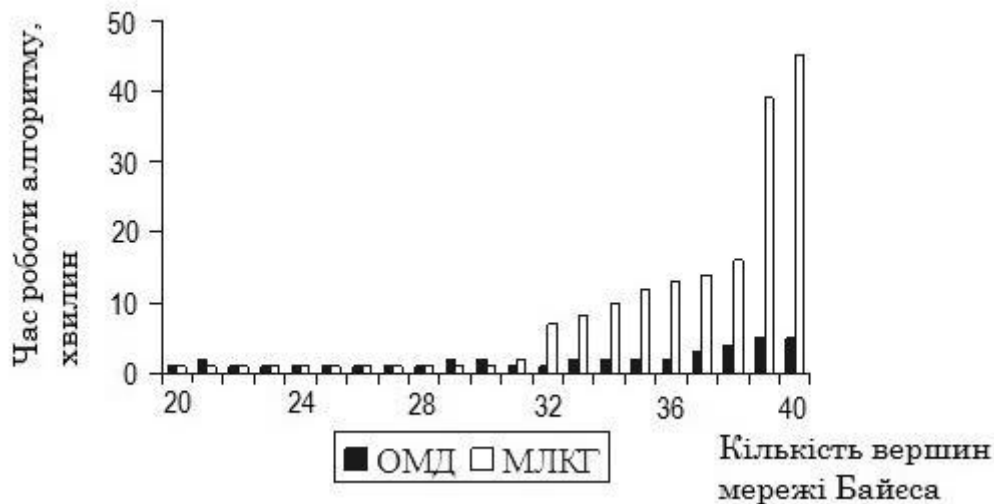


Рис. 5.2. Діаграма витрат часу на побудову мережі Байєса за алгоритмами з використанням модифікованої функції Купера-Герсковича та опису мінімальною довжиною

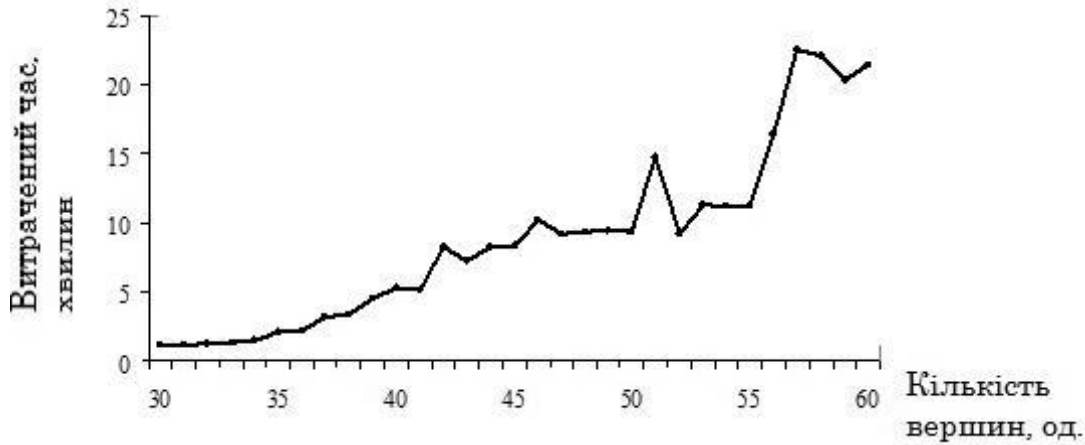


Рис. 5.3. Графік витрат часу при використанні алгоритму із застосуванням функції опису мінімальною довжиною

Для задавання порядку додавання дуг у мережі Байєса ці методи використовують значення взаємної інформації [173].

При побудові мережі Байєса в якості вибірки навчальних даних використано генетичні дані, які складаються з 600 навчальних записів. Як можна побачити з рис. 5.2 та 5.3, для побудови мереж Байєса, що складаються з більш ніж з 30 вершин, алгоритм із використанням опису мінімальною довжиною працює швидше, ніж модифікована функція Купера-Герсковича.

**5.7. Алгоритм евристичного метода побудови байєсівських мереж за навчальними даними**

Іншим прикладом побудови структури мережі Байєса є алгоритм евристичного методу.

**Вхідними даними** є множина навчальних даних  $D = \{d_1, \dots, d_n\}$ ,  $d_i = \{x_i^{(1)} x_i^{(2)} \dots x_i^{(N)}\}$  (нижній індекс – номер спостереження, а верхній – номер змінної);  $n$  – кількість спостережень;  $N$  – кількість вершин (змінних).

**Перший етап.** Для всіх пар вершин обчислюються значення взаємної інформації  $Set\_MI = \left\{ MI(x^i, x^j); \forall i, j \right\}$ .

Після цього елементи множини  $Set\_MI$  впорядковують за зменшенням:

$$Set\_MI = \{MI(x^{m_1}, x^{m_2}), MI(x^{m_3}, x^{m_4}), MI(x^{m_5}, x^{m_6}), \dots\}.$$

**Другий етап.**

**Крок 1.** З множини значень взаємної інформації  $Set\_MI$  обирають перші два максимальних значення  $MI(x^{m_1}, x^{m_2})$  та  $MI(x^{m_3}, x^{m_4})$ . За цими значеннями  $MI(x^{m_1}, x^{m_2})$  та  $MI(x^{m_3}, x^{m_4})$  будується множина моделей  $G$ :

$\{(m_1 \rightarrow m_2; m_3 \rightarrow m_4), (m_1 \rightarrow m_2; m_3 \leftarrow m_4), (m_1 \leftarrow m_2; m_3 \leftarrow m_4), (m_1 \leftarrow m_2; m_3 \rightarrow m_4), (m_1 \leftarrow m_2; m_3 \text{ не залежить від } m_4), (m_1 \rightarrow m_2; m_3 \text{ не залежить від } m_4), (m_1 \text{ не залежить від } m_2; m_3 \rightarrow m_4), (m_1 \text{ не залежить від } m_2; m_3 \leftarrow m_4), (m_1 \text{ не залежить від } m_2; m_3 \text{ не залежить від } m_4)\}$ .

Запис виду  $m_i \rightarrow m_j$  означає, що вершина  $x^{m_i}$  є предком вершини  $x^{m_j}$ .

**Крок 2.** Здійснюється пошук серед всіх моделей множини  $G$ . Параметр  $g^*$  зберігає оптимальну структуру мережі Байєса.

Оптимальною структурою буде та, у якій найменше значення функції  $L(g, x^n)$ , де  $L(g, x^n)$  – опис мінімальної довжини структури моделі при заданій послідовності з  $n$  спостережень  $x^n = d_1 d_2 \dots d_n$ .

*Алгоритм виконання кроку 2:*

1.  $g^* \leftarrow g_0 (\in G)$ ;
2. для  $\forall g \in G - \{g_0\}$  якщо  $L(g, x^n) < L(g^*, x^n)$  то  $g^* \leftarrow g$ ;
3. на вихід в якості рішення подається  $g^*$ .

**Крок 3.** Після того, як знайдена оптимальна структура (структури)  $g^*$  із  $G$ , з множини значень взаємної інформації  $Set\_MI$ , обирають наступне максимальне значення  $MI(x^{i\_next}, x^{j\_next})$ .

За отриманим значенням  $MI(x^{i\_next}, x^{j\_next})$  і структурою (структурами)  $g^*$  будується множина моделей  $G$  у вигляді:

$\{(g^*; i\_next \rightarrow j\_next), (g^*; i\_next \leftarrow j\_next), (g^*; i\_next \text{ не залежить від } j\_next)\}$ .

Після цього виконується крок 2.

**Умова завершення алгоритму.** Евристичний метод буде виконуватися до тих пір, доки не буде проаналізовано певну кількість елементів множини або всі  $\frac{N \cdot (N-1)}{2}$  елементи множини  $Set\_MI$ .

Як показує практика, у більшості випадків немає потреби виконувати аналіз більше половини (тобто  $\frac{N \cdot (N-1)}{4}$ ) елементів множини  $Set\_MI$ .

**Результат виконання алгоритму – побудована** оптимальна структура (структури)  $g^*$  мережі Байєса.

### **Приклад застосування евристичного методу**

В якості прикладу розглянемо загальновідому мережу "Asia", яка складається з восьми вершин. На рис. 5.4 наведена структура оригінальної мережі Байєса, на основі якої генерувалися значення.

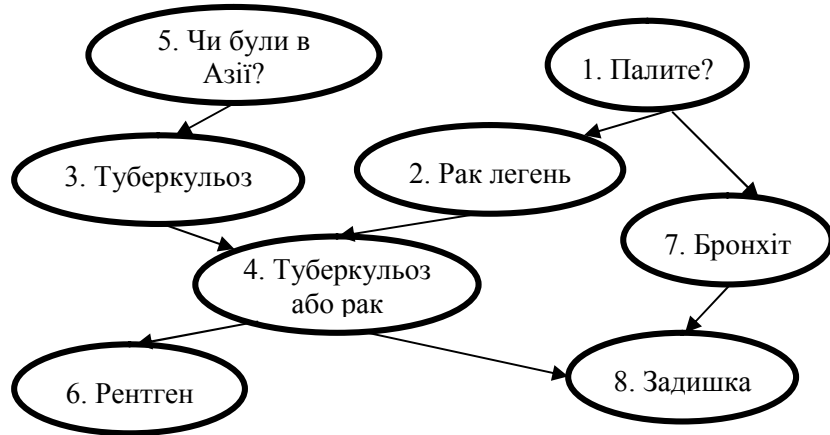


Рис. 5.4. Оригінальна мережа "Asia"

У табл. 5.11 наведені значення взаємної інформації між всіма вершинами мережі, розташовані за зменшенням (перший етап алгоритму).

Таблиця 5.11

Значення взаємної інформації між всіма вершинами мережі Байєса "Азія"

№	MI	i	j	№	MI	i	j
1	0,251	7	8	15	0,001227	3	5
2	0,136	2	4	16	0,000851	1	6
3	0,125	4	6	17	0,000508	2	7
4	0,096	2	6	18	0,000381	3	7
5	0,048	1	7	19	0,000266	4	5
6	0,036	3	4	20	0,000197	1	5
7	0,025	3	6	21	0,000128	4	7
8	0,0245	1	8	22	0,00012271	2	5
9	0,0132	4	8	23	0,00006475	5	6
10	0,0101	2	8	24	0,00003950	2	3
11	0,0051	6	8	25	0,00003249	5	7
12	0,0031	1	2	26	0,00001725	5	8
13	0,0028	3	8	27	0,00000303	1	3
14	0,0022	1	4	28	0,00000074	6	7

Побудова мережі виконувалася за вибіркою із 7000 навчальних спостережень. На 1-й ітерації за двома першими рядками  $MI(7,8)$  та  $MI(2,4)$ , відсортованої матриці  $MI$ , будується множина моделей з 9 структур. Після перебору цієї множини з'ясується, що мінімальне значення функції опису мінімальної довжини відповідає двом структурам (рис. 5.5).

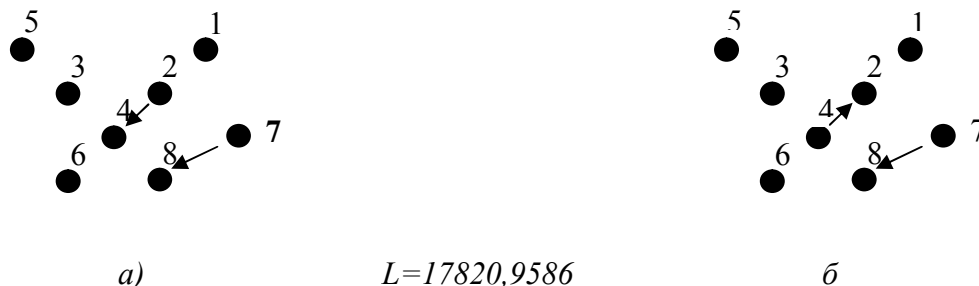


Рис. 5.5. Структури мережі Байєса, отримані на 1-й ітерації

На 2-й ітерації за отриманими на 1-й ітерації структурами (рис. 5.5) і  $MI(4,6)$  будеться множина моделей з 6 структур, для яких мінімальне значення функції опису мінімальної довжини одержує мережа, зображена на рис. 5.6.

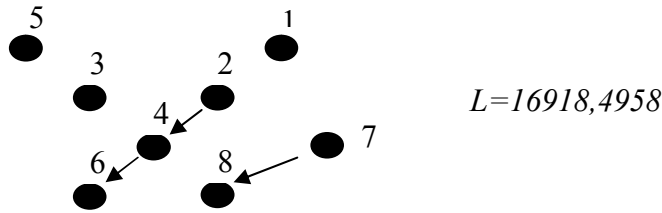


Рис. 5.6. Структура мережі Байєса, отримана на 2-й та 3-й ітераціях

На 3-й ітерації за отриманою на 2-й ітерацією структурою (рис. 5.6) та  $MI(2,6)$  будеться множина моделей з 3-х структур. З цих трьох структур мінімальне значення функції опису мінімальної довжини набуває та ж структура, що зображена на рис. 5.6.

На 4-й ітерації, за отриманою на 3-й ітерації структурою (рис. 5.6) і  $MI(1,7)$  будеться множина моделей з 3-х структур, для яких мінімальне значення функції опису мінімальної довжини набувають мережі, зображені на рис. 5.7.

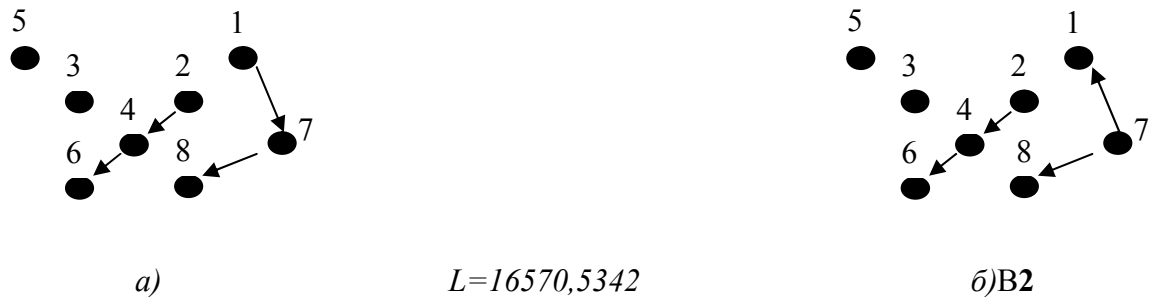


Рис. 5.7. Структури мережі Байєса, отримані на 4-й ітерації

На 5-й ітерації за структурами, отриманими на 4-й ітерації (рис. 5.7) і  $MI(3,4)$ , будеться множина моделей з 6-и структур, для яких мінімальне значення функції опису мінімальної довжини набувають мережі, зображені на рис. 5.8.

На 6-й ітерації за структурами, отриманими на 5-й ітерації і  $MI(3,6)$ , будеться множина моделей з 6-и структур. У результаті одержані ті ж самі структури, що і на 5-й ітерації (рис. 5.8).

На 7-й ітерації за структурами, отриманими на 6-й ітерації і  $MI(1,8)$ , будеться множина моделей з 6-и структур. Результат збігається з результатом попередніх двох ітерацій (рис. 5.8).

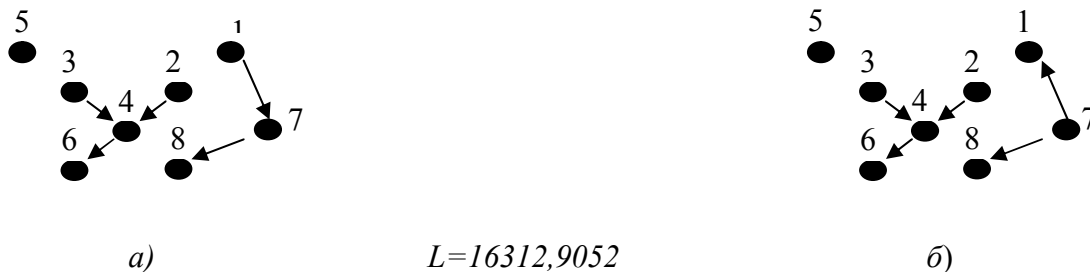


Рис. 5.8. Структури мережі Байєса, отримані на 5-й, 6-й та 7-й ітераціях

На 8-й ітерації за структурами, отриманими на 7-й ітерації (рис. 3.8) і  $MI(4,8)$ , будується множина моделей з 6-и структур, з яких мінімальне значення функції опису мінімальної довжини приймають мережі, зображені на рис. 5.9.

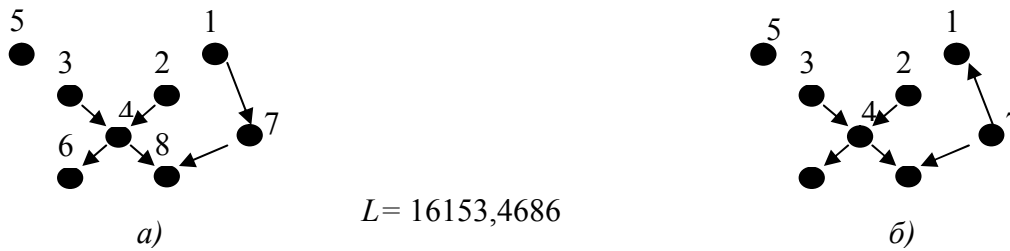


Рис. 5.9. Структури мережі Байєса, отримані на 8-й, 9-й та 10-й ітераціях

На 9-й ітерації за структурами, отриманими на 8-й ітерації і  $MI(2,8)$ , будується множина моделей з 6-и структур. У результаті одержані ті ж самі структури що і на 8-й ітерації (рис. 5.9).

На 10-й ітерації за структурами, отриманими на 9-й ітерації і  $MI(6,8)$ , будується множина моделей з 6-и структур. Результат збігається з результатом попередніх двох ітерацій (рис. 5.9).

На 11-й ітерації за структурами, отриманими на 10-й ітерації (рис. 5.9) і  $MI(1,2)$ , будується множина моделей з 6-и структур, з яких мінімальне значення функції опису мінімальної довжини набувають мережі, зображені на рис. 5.10. На 12-й ітерації за структурами, отриманими на 11-й ітерації і  $MI(3,8)$ , будується множина моделей з 6-ти структур.

У результаті одержані ті ж самі структури, що і на 11-й ітерації (рис. 5.10). На 13-й ітерації за структурами, отриманими на 12-й ітерації і  $MI(1,4)$ , будується множина моделей з 6-и структур. Результат збігається з результатом попередніх двох ітерацій (рис. 5.10).

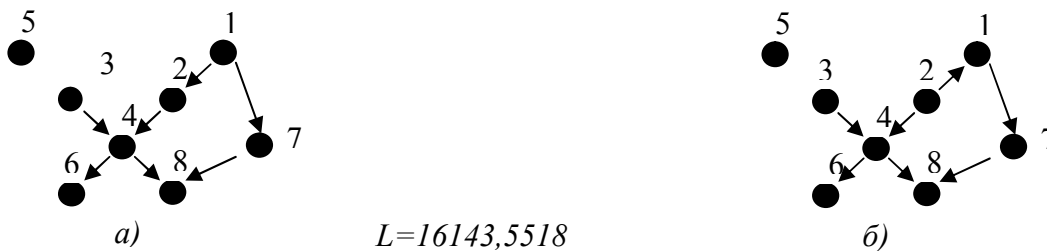


Рис. 5.10. Структури мережі Байєса, отримані на 11-й, 12-й та 13-й ітераціях

На 14-й ітерації за отриманими на 13-й ітерації структурами (рис. 5.10) і  $MI(3,5)$  будується множина моделей з 6-и структур, з яких мінімальне значення функції опису мінімальної довжини одержує мережа, зображена на рис. 5.11.

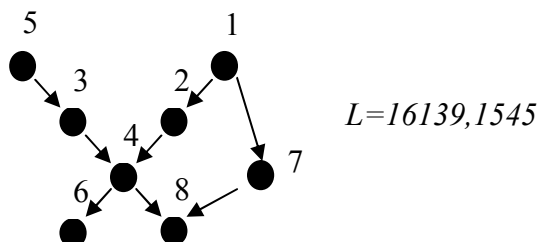


Рис. 5.11. Структура мережі Байєса, отримана з 14 по 27 ітерації

З 15-й по 27-й ітерації ніяких змін в структурі не відбувається, тобто результат збігається з результатом 14-й ітерації (рис. 5.11).

Для побудови мережі Байєса "Asia" при простому аналізі всіх можливих нециклічних структур буде потрібно виконати оцінювання 783702329343 моделей. Тоді як евристичний метод на 27-ми ітераціях алгоритму виконує аналіз усього лише 120 структур, причому вже на 14-й ітерації, після аналізу 81 структури метод видає структуру, що повністю збігається з оригінальною мережею "Asia".

Тобто наступні 13 ітерацій методу не роблять ніяких змін тому, що оптимальна структура знайдена вже на 14-й ітерації.

**Аналіз експериментальних результатів**

Проаналізуємо результати роботи алгоритму евристичного методу побудови байєсівських мереж за навчальними даними. Загалом для даного прикладу виконано шість обчислювальних експериментів. У кожному експерименті евристичним методом проводилося навчання мережі з 10 вершин за вибіркою з 2000 навчальних спостережень.

Для оцінювання якості навчання використано структурну різницю між побудованою і оригінальною мережами Байєса. Результати експериментів представлені на рис. 5.12-5.17.

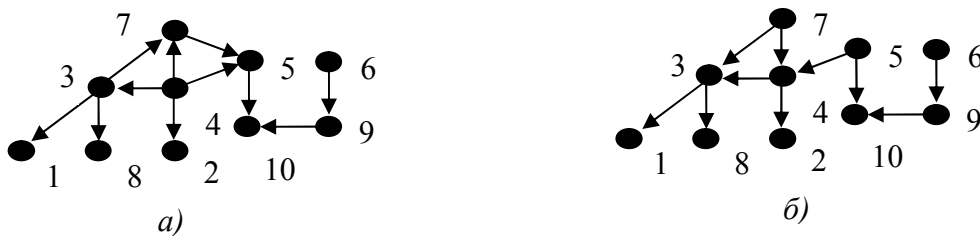


Рис. 5.12. Експеримент № 1 (а) побудована (б) оригінальна мережа Байєса

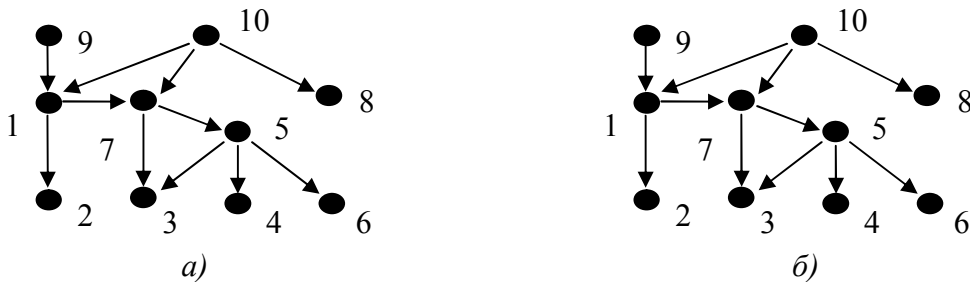


Рис. 5.13. Експеримент № 2 (а) побудована (б) оригінальна мережа Байєса

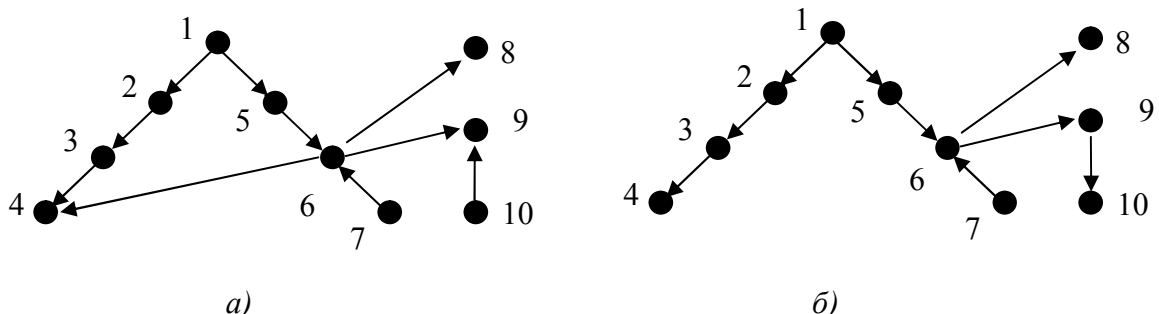


Рис. 5.14. Експеримент № 3 (а) побудована (б) оригінальна мережа Байєса

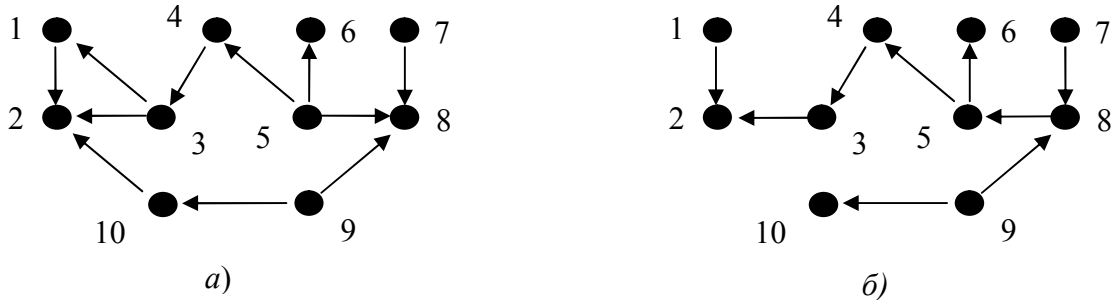


Рис. 5.15. Експеримент № 4 (а) побудована (б) оригінальна мережа Байєса

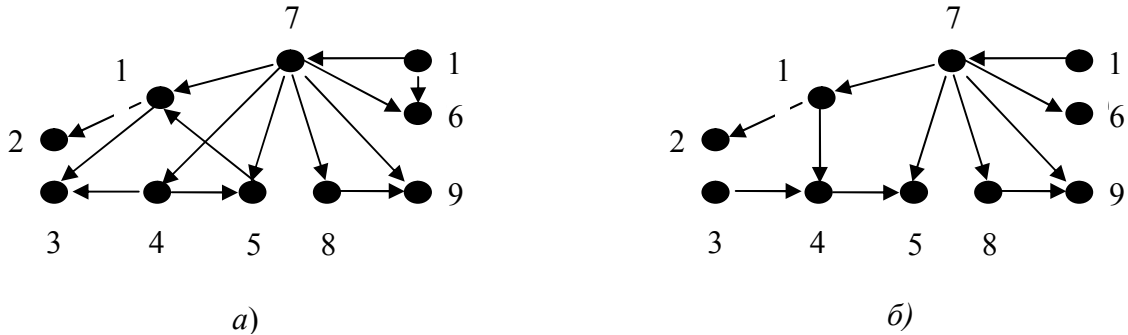


Рис. 5.16. Експеримент № 5 (а) побудована (б) оригінальна структура мережі

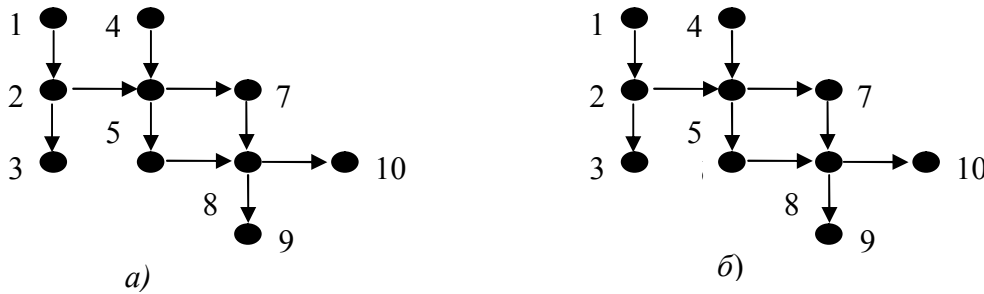


Рис. 5.17. Експеримент № 6 (а) побудована (б) оригінальна структура мережі Байєса

У табл. 5.12 згруповано результати шести обчислювальних експериментів. Для кожного експерименту виконано 44 ітерації навчання.

Таблиця 5.12

Результати шести обчислювальних експериментів

Номер обчислювального експерименту	№1	№2	№3	№4	№5	№6
Номер рисунку, який відповідає обчислювальному експерименту	рис. 5.12	рис. 5.13	рис. 5.14	рис. 5.15	рис. 5.16	рис. 5.17
Загальна кількість моделей, проаналізованих евристичним методом на всіх ітераціях	513	178	415	282	550	329
Зайві дуги	1	0	1	2	4	0
Відсутні дуги	0	0	0	0	1	0
Реверсовані дуги	3	0	1	1	1	0
Структурна різниця між побудованою та оригінальною мережею Байєса	8	0	3	4	7	0



Як можна бачити з табл. 5.12, у двох із шести обчислювальних експериментах № 2 та № 6 (рис. 5.13 та 5.17) побудована мережа повністю збіглася з оригінальною мережею Байєса. У двох із шести експериментах № 3 та № 4 (рис. 5.14 та 5.15) похибка навчання, тобто структурна різниця між побудованою та оригінальною моделями склала 3 і 4; для мережі з 10 вершин така похибка є цілком прийнятною.

Значні похибки навчання отримані в експериментах № 1 та № 5 (рис. 5.12 та 5.16). Однак для побудови мережі був виконаний аналіз усього лише 513 та 550 моделей, відповідно на всіх 44-х ітераціях, у той час, як при простому переборі, тобто аналізі “в лоб”, потрібно було проаналізувати 4 175 098 976 430 598 100 ациклічних моделей.

### 5.8. Методи оцінювання якості побудови байєсівських мереж

Для оцінювання якості побудови мережі Байєса можна використовувати облік кількості зайвих, відсутніх і реверсованих дуг у побудованій мережі Байєса порівняно з оригінальною мережею Байєса (мережею, за якою генерувалися дані). А в якості міри оцінювання похибки побудови можна використовувати *структурну різницю* (structure difference) або *перехресну ентропію* (cross entropy) між побудованою і оригінальною мережею Байєса.

#### Структурна різниця

Для обчислення структурної різниці використовують формулу симетричної різниці структур [176]:

$$\delta = \sum_{i=1}^n \delta_i = \sum_{i=1}^n \text{card}(\Pi^{(i)}(B) \Delta \Pi^{(i)}(A)) = \sum_{i=1}^n \text{card}((\Pi^{(i)}(B) \setminus \Pi^{(i)}(A)) \cup (\Pi^{(i)}(A) \setminus \Pi^{(i)}(B))), \quad (5.20)$$

де  $B$  – побудована мережа Байєса;  $A$  – оригінальна мережа Байєса;  $n$  – кількість вершин мережі;  $\Pi^{(i)}(B)$  – множина предків  $i$ -ї вершини побудованої мережі  $B$ ;  $\Pi^{(i)}(A)$  – множина предків  $i$ -ї вершини оригінальної мережі  $A$ ;  $\text{card}(\xi)$  – потужність скінченної множини  $\xi$ , що визначається кількістю елементів, які належать множині  $\xi$ .

#### Перехресна ентропія

Перехресна ентропія – це відстань між розподілом побудованої і оригінальної мережі Байєса. Нехай  $p(v)$  – спільний розподіл оригінальної мережі Байєса, а  $q(v)$  – спільний розподіл побудованої мережі Байєса.

Тоді перехресна ентропія обчислюється так [150]:

$$\begin{aligned} H(p, q) &= \sum_v p(v) \cdot \log \frac{p(v)}{q(v)} = \\ &= \sum_{j \in J} \sum_{s \in S(j, g)} \sum_{a \in A^{(j)}} p(X^{(j)} = a | \Pi^{(j)} = s) \cdot \log \frac{p(X^{(j)} = a | \Pi^{(j)} = s)}{q(X^{(j)} = a | \Pi^{(j)} = s)}. \end{aligned} \quad (5.21)$$

Наприклад, структурна різниця між мережами, зображеними на рис. 5.18, буде дорівнювати восьми.

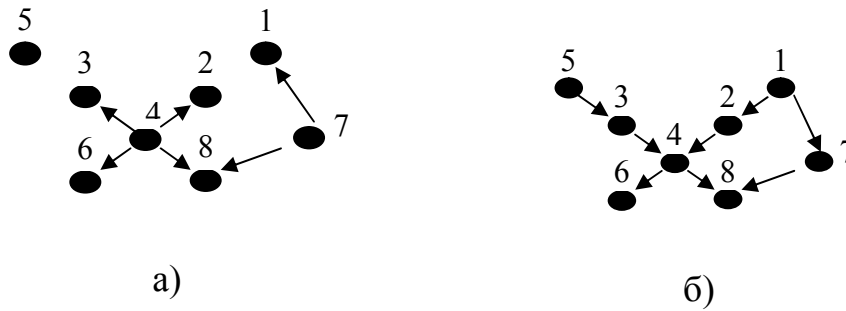


Рис. 5.18. Структури мережі Байєса для прикладу обчислення

Проміжні значення, які застосовуються для обчислення структурної різниці, наведені в табл. 5.13.структурної різниці

Таблиця 5.13

Проміжні розрахункові значення структурної різниці між мережами, зображеними на рис. 5.18

Номер вершини	Множина батьків вершини структури		$\Pi^{(i)}(B)\Delta\Pi^{(i)}(A)$	$card (\Pi^{(i)}(A)\Delta\Pi^{(i)}(B))$
	рис. 5.18.а	рис. 5.18.б		
1	$\Pi^{(1)} = \{7\}$	$\Pi^{(1)} = \{ \}$	$\{7\}$	1
2	$\Pi^{(2)} = \{4\}$	$\Pi^{(2)} = \{1\}$	$\{1,4\}$	2
3	$\Pi^{(3)} = \{4\}$	$\Pi^{(3)} = \{5\}$	$\{4,5\}$	2
4	$\Pi^{(4)} = \{ \}$	$\Pi^{(4)} = \{2,3\}$	$\{2,3\}$	2
5	$\Pi^{(5)} = \{ \}$	$\Pi^{(5)} = \{ \}$	$\{ \}$	0
6	$\Pi^{(6)} = \{4\}$	$\Pi^{(6)} = \{4\}$	$\{ \}$	0
7	$\Pi^{(7)} = \{ \}$	$\Pi^{(7)} = \{1\}$	$\{1\}$	1
8	$\Pi^{(8)} = \{4,7\}$	$\Pi^{(8)} = \{4,7\}$	$\{ \}$	0

З результатів, наведених обчислювальних експериментів, можна зробити висновок, що у більшості випадків похибка навчання за евристичним методом є прийнятною, а економія обчислювальних ресурсів і часу дуже великою.

Використання евристичного методу побудови мереж істотно розширює можливості використання мереж Байєса при виконанні аналізу в різних областях практичної діяльності, особливо там, де доводиться працювати з великими обсягами інформації.

Результати оцінювання складності алгоритмів навчання мереж Байєса згруповано у таблиці 5.14.

Таблиця 5.14

Результати оцінювання складності алгоритмів навчання мереж Байєса

№	Назва алгоритму	Складність
1	$K2$	$O(N^3)$
2	$K3$	$O(N^3)$
3	Модифікований $K3$	$O(N^4/2)$
4	$FBC$	$O(N^2)$
5	$KDB$	$O(N^2)$
6	Алгоритм Чу і Ліу	$O(N^2)$
7	Евристичний методів з використанням значень взаємної інформації та опису мінімальної довжини	$O(N^2/2)$

### Контрольні питання

1. Дайте означення та наведіть приклад обчислення значення взаємної інформації. Наведіть приклади інших мір зв'язку між змінними.
2. Дайте означення структури математичної моделі.
3. Що ви розумієте під структурою мережі Байєса?
4. Назвіть типи існуючих методів побудови структури мережі Байєса.
5. Міра Кулбака-Леблера – формула для обчислення та її зміст.
6. Чим зумовлена нелінійна поліноміальна складність задачі побудови структури мережі Байєса?
7. Для чого використовують рекурсивну формулу Робінсона?
8. Яка роль алгоритму максимізації математичного очікування при побудові структури мережі Байєса?
9. Що означає термін «прихована змінна»? Чи враховуються приховані змінні при побудові мережі Байєса?
10. Поясніть суть коефіцієнтів асоціації та контингенції.
11. Які методи на основі оціночних функцій існують для побудови мереж Байєса? Наведіть приклади.
12. Що таке ентропія і як її можна використати при побудові структури мережі Байєса?
13. Які методи із використанням тесту на умовну незалежність існують для побудови мереж Байєса? Наведіть приклади.
14. Які статистичні показники зв'язку між змінними використовуються для побудови мереж Байєса? Наведіть приклади.
15. Для чого використовується метод упорядкування множини вершин? Наведіть приклад обчислення.
16. Формула та зміст функції Купера-Герсковича: у чому її недолік та як його можна подолати? Наведіть приклади використання.
17. Який вигляд має функція опису мінімальної довжини? Наведіть означення, формулу, приклад.

18. Що означає слово «евристичний»?
19. Розгляньте алгоритм евристичного методу побудови мережі Байєса на основі використання значень взаємної інформації та функції опису мінімальної довжини.
20. Які існують методи оцінювання якості побудованої мережі Байєса. Наведіть формули структурної різниці та перехресної ентропії.
21. У чому полягає різниця між евристичними та точними методами побудови мереж Байєса?

## РОЗДІЛ 6

### ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ МЕРЕЖ БАЙЄСА

- Означення архітектури СППР та її компоненти
- Обов'язкові елементи СППР з точки зору системного підходу
- Системи з фіксованим і гнучким використанням обчислювальних процедур, комбінованого (гібридного) типу
- Докладний розгляд процесу розробки та реалізації інформаційної системи підтримки прийняття рішень, створення прототипу СППР

#### 6.1. Загальна характеристика систем підтримки прийняття рішень

Архітектура СППР – узагальнена структурна схема системи, яка відображає її основні функції, зв'язки між підсистемами та можливості стосовно розв'язання конкретних задач. Приклад простої архітектури СППР для прогнозування альтернативними методами представлено на рис. 6.1.

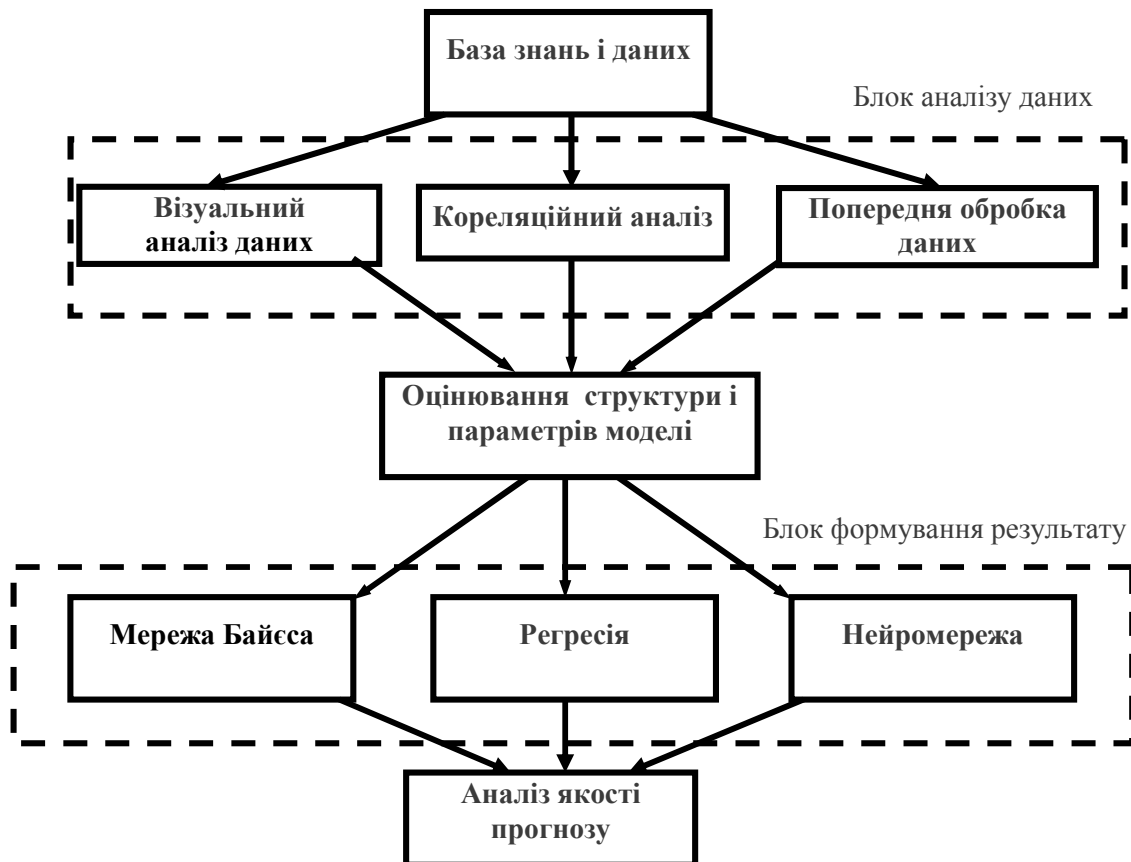


Рис. 6.1. Архітектура системи підтримки прийняття рішень

Методи попередньої обробки даних застосовують для встановлення зв'язків між змінними та формування структури моделі на основі отриманої інформації. Структура моделі може будуватись (оцінюватись) для байєсівської мережі, регресійної моделі або для нейромережі. Моделі двох останніх типів доповнюють основний результат, який обчислюється за допомогою мережі Байєса.

У всіх випадках необхідно визначати елементи структури моделі того чи іншого типу.

**У загальному випадку система підтримки прийняття рішень складається з таких компонент:**

- мовна система;
- система попередньої обробки даних та генерування результатів;
- база знань і даних;
- система представлення проміжних та остаточних результатів.

Узагальнена концептуальна модель (структура) системи підтримки прийняття рішень представлена на рисунку 6.2.

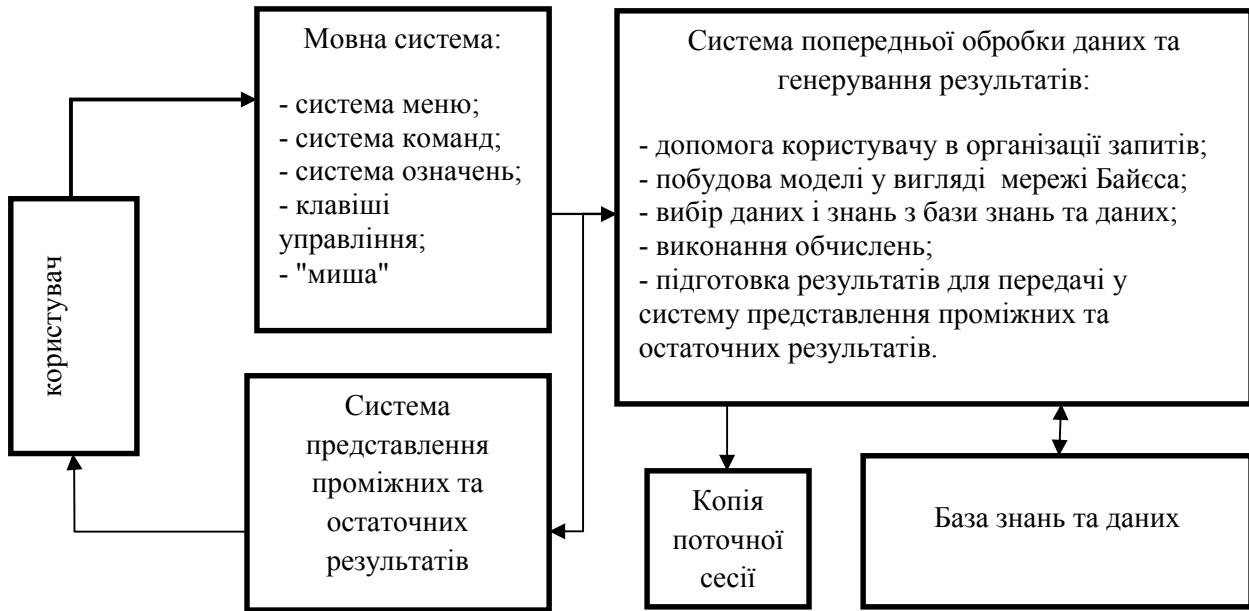


Рис. 6.2. Узагальнена концептуальна модель системи підтримки прийняття рішень

### **Компоненти систем підтримки прийняття рішень**

**Мовна система** – це сукупність мовних конструкцій, які включають всі запити, які може зробити користувач.

Мовна система може бути реалізована на основі:

- використання командних рядків;
- меню;
- графічних елементів (іконок);
- натуральних мовних конструкцій;
- заповнення стандартизованих форм;

- на основі маніпулювання безпосередньо об'єктом проектування чи дослідження у графічному вигляді.

**Система попередньої обробки даних і генерування результатів** – це основна система підтримки прийняття рішень, яка реагує на запити користувача, виконує задані дії і генерує належну відповідь на запит.

Система попередньої обробки даних і генерування результатів приймає запити користувача, аналізує їх коректність, вибирає необхідні знання і дані з бази знань та даних, виконує необхідні обчислювальні дії і передає результати в систему представлення результатів. У процесі функціонування системи попередньої обробки

даних і генерування результатів може розширювати або модифікувати базу знань та даних.

**База знань і даних** містить всі необхідні типи знань і даних для конкретного застосування.

Зміст бази знань і даних:

- строго структуровані знання у вигляді записів;
- лінгвістичні знання – лінгвістичні правила, змінні;
- числові описові знання (константи);
- процедурні знання – обчислювальні алгоритми;
- логічні правила щодо вибору варіантів моделей, прогнозів і т. ін.;
- знання (форми) щодо представлення результатів – таблиці, діаграми, графіки; наприклад, методи формування ймовірнісних висновків за мережею, композиційне правило формування висновку, дерево рішень;
- асимілюючі знання – знання стосовно того, як доповнити чи розширити базу знань та даних.

**Система представлення результатів** – це сукупність мовних і графічних конструкцій, які містять всі типи повідомлень користувачу, які може згенерувати система підтримки прийняття рішень.

Вибір елемента, що використовується як повідомлення, виконує система попередньої обробки даних і генерування результатів. Елементом може бути текст, таблиця, графік і т. ін.

**Послідовність роботи з системою підтримки прийняття рішень**

1. Користувач дає запит до системи підтримки прийняття рішень за допомогою елементів мовної системи. Характерними запитами є такі:

- запит на введення в базу знань і даних нових даних і знань;
- запит на розв'язання конкретної задачі;
- запит на уточнення попереднього запиту;
- запит щодо надання допомоги.

2. Запит передається в систему попередньої обробки даних і генерування результатів, яка виконує його обробку. З бази даних і знань вибираються необхідні дані і знання для виконання запиту, виконується обробка даних, користувачу видається повідомлення про виконання запиту і результати передаються в систему представлення проміжних та остаточних результатів.

3. Система представлення проміжних та остаточних результатів представляє результати у заданому форматі.

**Системний підхід до проектування інформаційних систем підтримки прийняття рішень** передбачає виконання таких вимог:

- наявність альтернативних (взаємодоповнюючих) методів обробки даних, які забезпечують збереження та підвищення інформативності даних, а також приведення їх до форми, зручної для виконання подальших обчислень (підвищення інформативності може бути досягнуто за рахунок розширення об'єму вибірки, корегування типу розподілу (часто перехід від рівномірного до нормального розподілу сприяє підвищенню інформативності даних), застосування методів імітаційного моделювання);

- наявність альтернативних методів оцінювання структури і параметрів математичних і статистичних моделей – методи оптимізації структури мережі

Байєса та оцінювання параметрів лінійних і нелінійних моделей;

- використання множини критеріїв оцінювання ступеня адекватності побудованої моделі;
- наявність альтернативних методів генерування ймовірнісних висновків та (можливо) керуючих впливів;
- застосування множини критеріїв для оцінювання якості ймовірнісного висновку, оцінок прогнозів та керуючих впливів;
- правила вибору кращого рішення з множини згенерованих альтернатив, тобто метод формування остаточного висновку.

### 6.2. Особливості систем підтримки прийняття рішень на основі мереж Байєса різної типології

Системи підтримки прийняття рішень на основі мереж Байєса відносяться до систем, що ґрунтуються на розрахункових процедурах. Тобто система такого типу містить множини обчислювальних алгоритмів для розв'язання задач попередньої обробки даних, оцінювання структури і параметрів мережі, а також для формування ймовірнісного висновку. Як правило, процедури фокусуються на задачах таких класів, як: прогнозування, планування, поглиблений статистичний аналіз даних, розв'язання оптимізаційних задач та інші.

Існують два основні підходи щодо використання обчислювальних алгоритмів у системах підтримки прийняття рішень: фіксований та гнучкий. При фіксованому підході алгоритм є частиною системи обробки даних та генерації результатів, а це означає, що до системи підтримки прийняття рішень не можна легко додати / видалити нові алгоритми або модифікувати її в цілому. Розглянемо докладніше систему з фіксованим використанням алгоритмів (рис. 6.3).

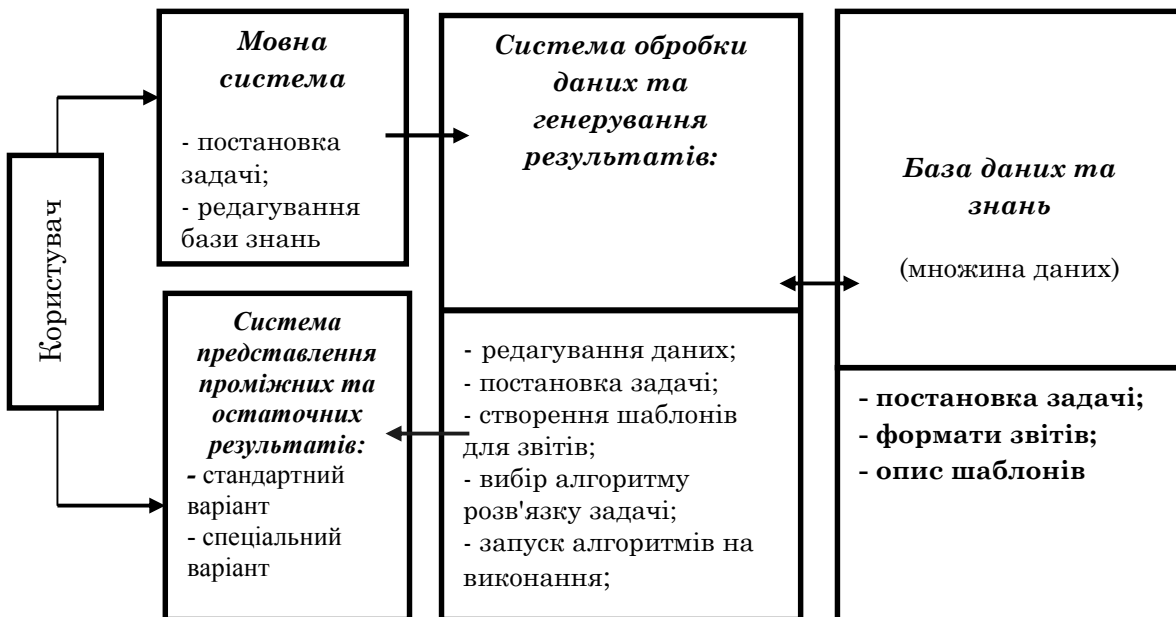


Рис. 6.3. Структура системи підтримки прийняття рішень з фіксованим використанням алгоритмів



При використанні *гнучкого підходу* можна легко добавляти, видаляти, модифікувати та об'єднувати множину алгоритмів у процесі експлуатації системи підтримки прийняття рішень. При цьому база знань та даних може містити дані, постановки задач, формати звітів, правила прийняття рішень з метою координації виконання алгоритмічних модулів.

Відмінність гнучкої системи від жорсткої полягає в тому, що база знань містить алгоритмічні модулі, які можна комбінувати в необхідній послідовності та запускати на виконання. Структура системи підтримки прийняття рішень гнучкого типу на основі обчислювальних алгоритмічних процедур наведена на рисунку 6.4.



Рис. 6.4. Структура системи підтримки прийняття рішень гнучкого типу на основі обчислювальних алгоритмічних процедур

Якщо системи підтримки прийняття рішень на основі алгоритмічних процедур об'єднати, наприклад, з системою підтримки прийняття рішень на основі вирішальних правил, то отримуємо *комбіновану (гібридну) систему*. Часто об'єднують системи на основі мереж Байєса, статистичних розрахунків, нечіткої логіки, дерев рішень, нелінійних регресійних моделей і т. ін. Архітектура такої системи наведена на рис. 6.5.

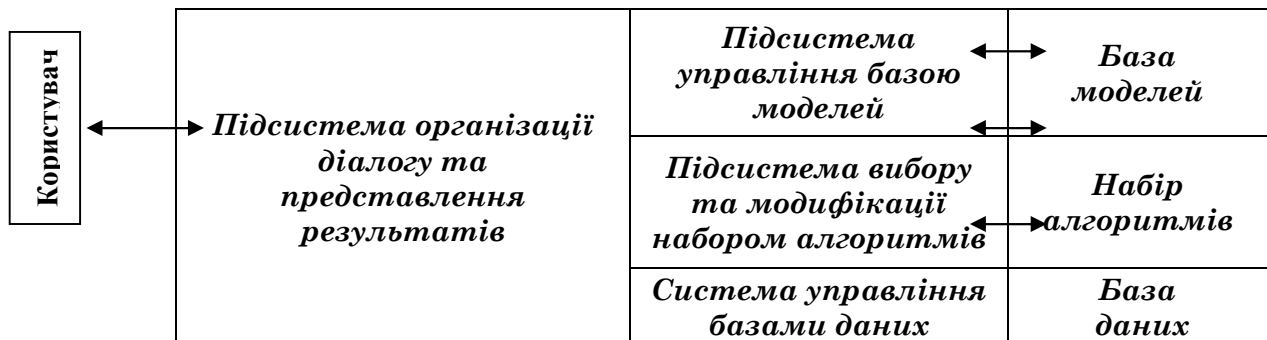


Рис. 6.5. Структура системи підтримки прийняття рішень комбінованого (гібридного) типу

Безперечно, що такі системи є більш універсальними та дають більше можливостей з точки зору запитів і отриманих результатів. При цьому система підтримки прийняття рішень може сама приймати рішення про те, який метод обробки даних і знань вибирати в конкретному випадку. Можливим запитом до такої системи є: “показати середній приріст промислового виробництва для південних районів України.” На рис. 6.6 наведена альтернативна архітектура подібної системи.

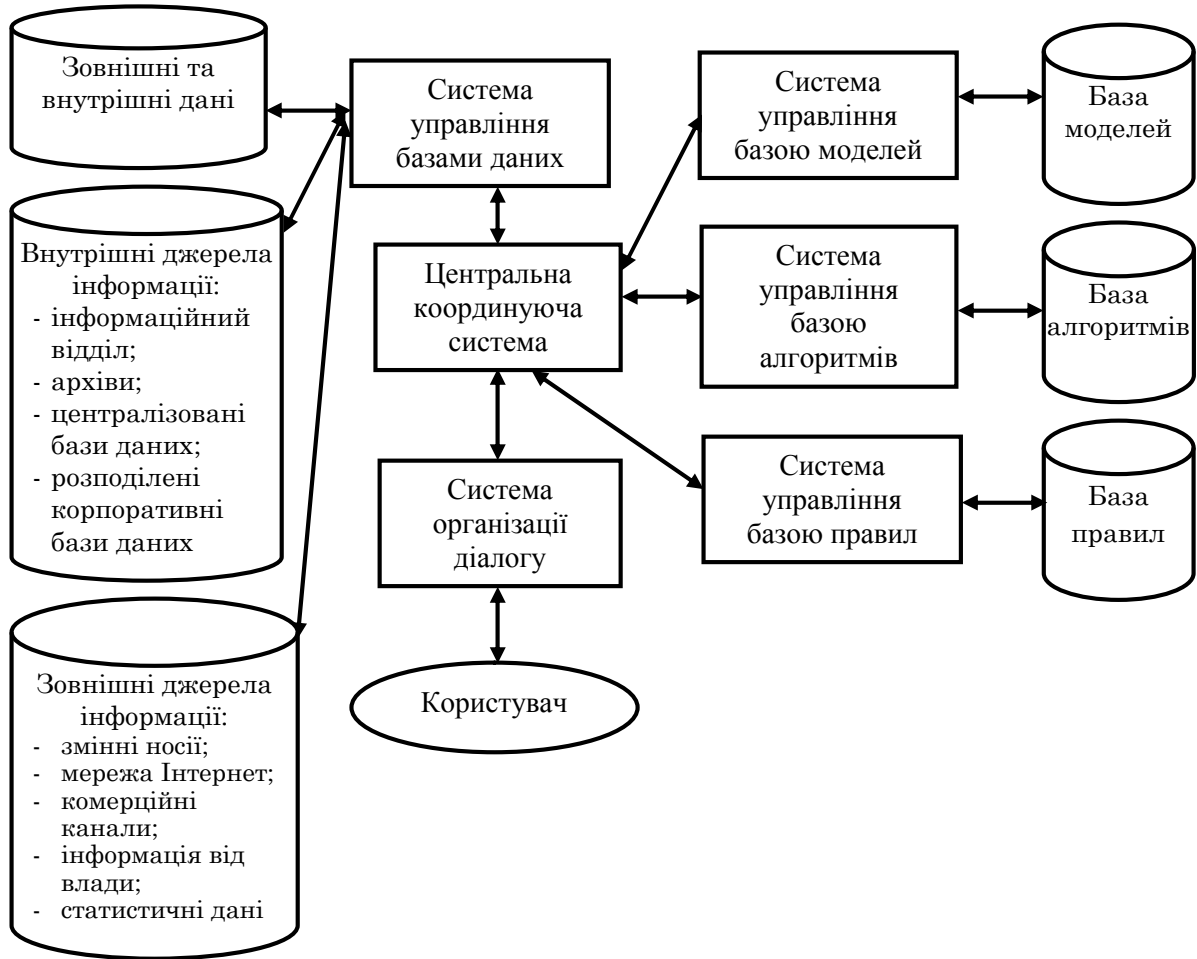


Рис. 6.6. Альтернативна архітектура системи комбінованого типу

Системи підтримки прийняття рішень комбінованого типу мають такі особливості:

Використання внутрішніх джерел інформації:

- інформаційний відділ;
- архіви;
- централізована база даних;
- розподілені корпоративні бази даних.

Наявність зовнішніх джерел інформації:

- інформація на CD ROM;
- Internet;
- комерційні канали;
- інформація від влади;
- інформація від статистичних органів.

Наприклад, за таким принципом побудовані системи підтримки прийняття рішень для управління повітряним рухом у великих аеропортах; вони забезпечують виконання таких основних функцій:

- чисто довідкові;
- розрахунок оптимальних траєкторій польоту в екстремальних ситуаціях;
- автоматизований гнучкий розклад посадок / зльоту;
- система підтримки прийняття рішень як радник для персоналу;
- стан парку літаків, приписаних до аеропорту;
- розв'язання задачі оптимального розподілу злітно-посадкових смуг.

### **6.3. Система підтримки прийняття рішень для інтелектуального аналізу даних на основі байєсівських мереж**

Одним із варіантів гібридних систем підтримки прийняття рішень є комп'ютерні системи, що спрямовані на розв'язання таких задач інтелектуального аналізу даних, як аналіз зв'язків, класифікація, кластеризація, прогнозування та візуалізація, що здійснюються без залучення експертів – за навчальними даними шляхом формування відповідного ймовірнісного висновку. Досить ефективними в даній галузі є системи підтримки прийняття рішень для інтелектуального аналізу даних на основі байєсівських мереж.

Перевагами систем підтримки прийняття рішень на основі мереж Байєса є гнучка архітектура, що передбачає введення нових алгоритмічних модулів для реалізації будь-яких методів побудови мереж Байєса та ймовірнісного висновку, або навіть інших методів інтелектуального аналізу даних.

За рівнем користувача система підтримки прийняття рішень інтелектуального аналізу даних на основі мереж Байєса належить до класу *активних систем підтримки прийняття рішень*, тобто ця система допомагає особі, що приймає рішення обрати найбільш прийнятне, користуючись значеннями ймовірностей виникнення тієї чи іншої події, а не тільки допомагає у прийнятті рішень шляхом виконання складних обчислювальних операцій [124].

За інструментальним підходом класифікації, розробленої Спрагом, система підтримки прийняття рішень для інтелектуального аналізу даних на основі мереж Байєса відноситься до *СППР-генератора*, тому що кожна мережа Байєса, яка побудована на основі навчальних даних, представляє собою причинно-наслідкову мережу, що може розглядатись як окрема система підтримки прийняття рішень [126].

Система підтримки прийняття рішень для інтелектуального аналізу даних на основі мереж Байєса складається з такого ж набору компонентів, що й інші системи такого типу: трьох основних підсистем і передбачає модульно-блочну побудову. До її складу входять блоки, що призначені для накопичення даних; побудови моделей (ймовірнісних моделей у вигляді мереж Байєса) та формування ймовірнісного висновку відносно заданого того чи іншого стану процесу.

*Пристрої вводу-виводу* надають користувачу можливість завантажувати дані в систему підтримки прийняття рішень. Для цього підсистема вводу-виводу функціонально зв'язана з підсистемою інтерфейсу користувача.

*Підсистема інтерфейсу користувача* призначена для здійснення зв'язку між користувачами системи підтримки прийняття рішень та внутрішніми елементами

системи і забезпечує ввід та вивід інформації для особи, що приймає рішення і експертів, а також надає доступ до зовнішніх запам'ятовуючих пристроїв ПЕОМ. Інтерфейс дозволяє операторові вводити інформацію, дані, команди, параметри і запити в систему та одержувати вихідну інформацію в зручному для сприйняття вигляді.

Для організації діалогу користувач–система в системі підтримки прийняття рішень реалізовані наступні запити *мовної системи*:

- на формування конкретних процедур обробки даних та прогнозування (формулювання вимог);
- на вибір та формулювання критеріїв розв'язання задачі;
- на виконання задач моделювання і прогнозування;
- на форму представлення результатів.

Головним з точки зору обробки даних у даній системі є модуль обробки даних та генерації результатів. Він сприймає коректні запити користувача і виконує наступні задачі:

- читання необхідних даних у вигляді таблиць навчальних даних для побудови моделей;
- вибір методів побудови мережі Байєса;
- запуск на виконання модулів обробки даних та прогнозування;
- формування результатів обробки даних та їх зберігання в короткостроковій та довгостроковій пам'яті;
- генерування діагностичних повідомлень.

*Підсистема зберігання інформації* складається з бази даних, бази моделей та бази знань, які призначені для накопичення даних, моделей у вигляді структур мережі Байєса, таблиць умовних ймовірностей та метаданих.

*Головна підсистема аналізу* складається з трьох блоків:

- обробки даних;
- побудови структури мережі Байєса;
- побудови ймовірнісного висновку.

Ця система призначена для аналізу даних від користувача системою, побудови мережі Байєса та ймовірнісного висновку за цими даними. Головна підсистема аналізу отримує дані по шині даних від підсистеми зберігання інформації.

*Блок обробки даних* призначений для перевірки даних, що поступають від головної підсистеми аналізу в блок обробки даних на наявність пропусків та викидів (екстремальних значень), а також для здійснення операції агрегування над цими даними.

Блок обробки даних складається з чотирьох модулів:

- аналізу неповних даних;
- видалення записів з неповними даними;
- заповнення неповних даних;
- агрегування даних.

*Модуль аналізу повноти даних* призначений для перевірки вхідних навчальних даних на наявність пропусків та викидів. Модуль аналізу повноти даних зв'язаний з модулями видалення записів з неповними даними, заповнення неповних даних, агрегування даних.

Якщо запис навчального набору даних, що аналізується, містить невелику кількість пропусків (відсутніх даних), то цей запис передається для заповнення

найбільш ймовірним значенням для даного запису в модуль заповнення неповних даних, інакше такий запис видаляється з множини навчальних даних в модулі видалення записів з неповними даними.

*Модуль видалення записів з неповними даними* призначений для видалення записів з навчальних даних, що мають велику кількість пропусків.

*Модуль заповнення неповних даних* призначений для заповнення записів з навчальних даних найбільш ймовірним значенням, для цього пропонується використовувати спрощену („наївну”) мережу Байєса.

*Модуль агрегування даних* призначений для виконання операції агрегування даних відносно множини навчальних даних. Операція агрегування здійснюється наступним чином: якщо якийсь вузол з множини навчальних даних приймає числові значення, то ці числові значення замінюються інтервальними оцінками.

*Блок побудови структури мережі Байєса* складається з таких модулів:

- обчислення тесту на умовну незалежність;
- пошуку оптимальної структури мережі Байєса.

Цей блок призначений для побудови структури мережі Байєса за даними, що передаються з блока обробки даних.

*Модуль обчислення тесту на умовну незалежність* призначений для обчислення матриці взаємної залежності між вершинами (факторами процесу). У системі підтримки прийняття рішень для інтелектуального аналізу даних на основі мереж Байєса в якості тесту на умовну незалежність використовується значення взаємної інформації.

*Модуль пошуку оптимальної структури мережі Байєса* реалізує цілеспрямований пошук оптимальної структури мережі, в якості критерію оптимальності застосовується функція опису мінімальною довжиною.

*Блок побудови ймовірнісного висновку* призначений для побудови ймовірнісного висновку по структурі мережі Байєса, що надходить з блока побудови її структури. Даний блок складається з чотирьох модулів:

- перевірки наявності інстанційованих вершин;
- побудови таблиці умовних ймовірностей;
- побудови матриці емпіричних значень спільного розподілу ймовірностей всієї мережі;
- побудови ймовірнісного висновку.

*Модуль перевірки наявності інстанційованих вершин* використовується, якщо множина інстанційованих значень порожня, то система підтримки прийняття рішень передає керування в модуль побудови таблиці умовних ймовірностей, інакше модулю побудови матриці емпіричних значень спільного розподілу ймовірностей всієї мережі.

*Модуль побудови таблиць умовних ймовірностей* призначений для побудови таблиць умовної ймовірності кожного вузла мережі Байєса за множиною навчальних даних.

*Модуль побудови матриці емпіричних значень спільного розподілу ймовірностей всієї мережі* призначений для побудови матриці емпіричних значень сумісного розподілу ймовірностей всієї мережі Байєса за навчальними даними.

*Модуль побудови ймовірнісного висновку.* Залежно від того, від якого модуля прийшов сигнал, виконуються відповідні обчислення. Якщо сигнал прийшов від модуля побудови таблиці умовних ймовірностей, то застосовується класичний метод

прямого розповсюдження ймовірностей по мережі Байєса. Спочатку обчислюються ймовірності значень кореневих вершин, тобто вершин, у яких відсутні батьківські вершини. Після цього визначаються ймовірності усіх інших вузлів за формулою і таблиця умовних ймовірностей. Якщо сигнал надійшов від модуля побудови матриці емпіричних значень спільного розподілу ймовірностей всієї мережі, то розрахунки виконуються у такий спосіб. На основі інформації про структуру моделі мережі Байєса та інстанційовані значення, виконується послідовний перебір всіх вузлів мережі. Якщо вузол не інстанційований, то виконується розрахунок усіх можливих станів цього вузла. Після цього виконується послідовний перебір усіх рядків матриці емпіричних значень спільного розподілу ймовірностей всієї мережі Байєса і, якщо значення вузла рядка співпадає із значенням інстанційованих вузлів і значеннями аналізованого вузла, то відповідне значення додається до значення ймовірності відповідного стану аналізованого вузла. Завершується опрацювання нормуванням значень ймовірностей станів аналізованого вузла.

На основі описаної архітектури можна реалізувати відповідне програмне забезпечення, використовуючи об'єктно-орієнтовані засоби програмування. Це дозволяє представити розроблені алгоритми у вигляді ієрархії класів з наслідуваними властивостями та методами, надаючи можливість одержати найбільш ефективну структуру алгоритмів.

### **6.4. Приклад практичної реалізації системи підтримки прийняття рішень для інтелектуального аналізу даних на основі мереж Байєса**

В якості прикладу практичного застосування методів та алгоритмів інтелектуального аналізу даних слід навести систему підтримки прийняття рішень для інтелектуального аналізу даних на основі байєсівських мереж, розроблену О. М. Терентьевим, що вирізняється гнучкою побудовою і передбачає функціонально-блочну архітектуру.

У запропонованій системі експерти не беруть участі у побудові моделей, оскільки розроблені методи побудови моделей та формування ймовірнісного висновку призначені саме для автоматичного аналізу процесів за даними, що їх описують. Дана система підтримки прийняття задовольняє основним характеристикам подібних систем: використовує дані і моделі у вигляді мереж Байєса; призначена для надання допомоги особі, що приймає рішення при прийнятті рішень для структурованих та неструктурованих задач; підтримує, а не замінює рішення, що приймаються особою, що приймає рішення; мета застосування створеної системи – підвищення якості та ефективності рішень.

Запропонована архітектура системи підтримки прийняття рішень стала основою для розробки комп'ютерного програмного забезпечення, яке за навчальними даними буде модель процесу у вигляді причино-наслідкової мережі Байєса. Програма була використана для дослідження процесів різної природи, а саме:

- побудови скорингової системи для оцінювання кредитоспроможності фізичних осіб у вигляді мережі Байєса для одного з українських банків;
- розробки моделі прогнозування композитного індексу нью-йоркської фондової біржі для українського підприємства – акціонерного товариства;

– медичної експертної системи прогнозування інфаркту міокарда у вигляді мережі Байєса; для навчання використана комп’ютерна база даних медичного університету Нью-Джерсі.

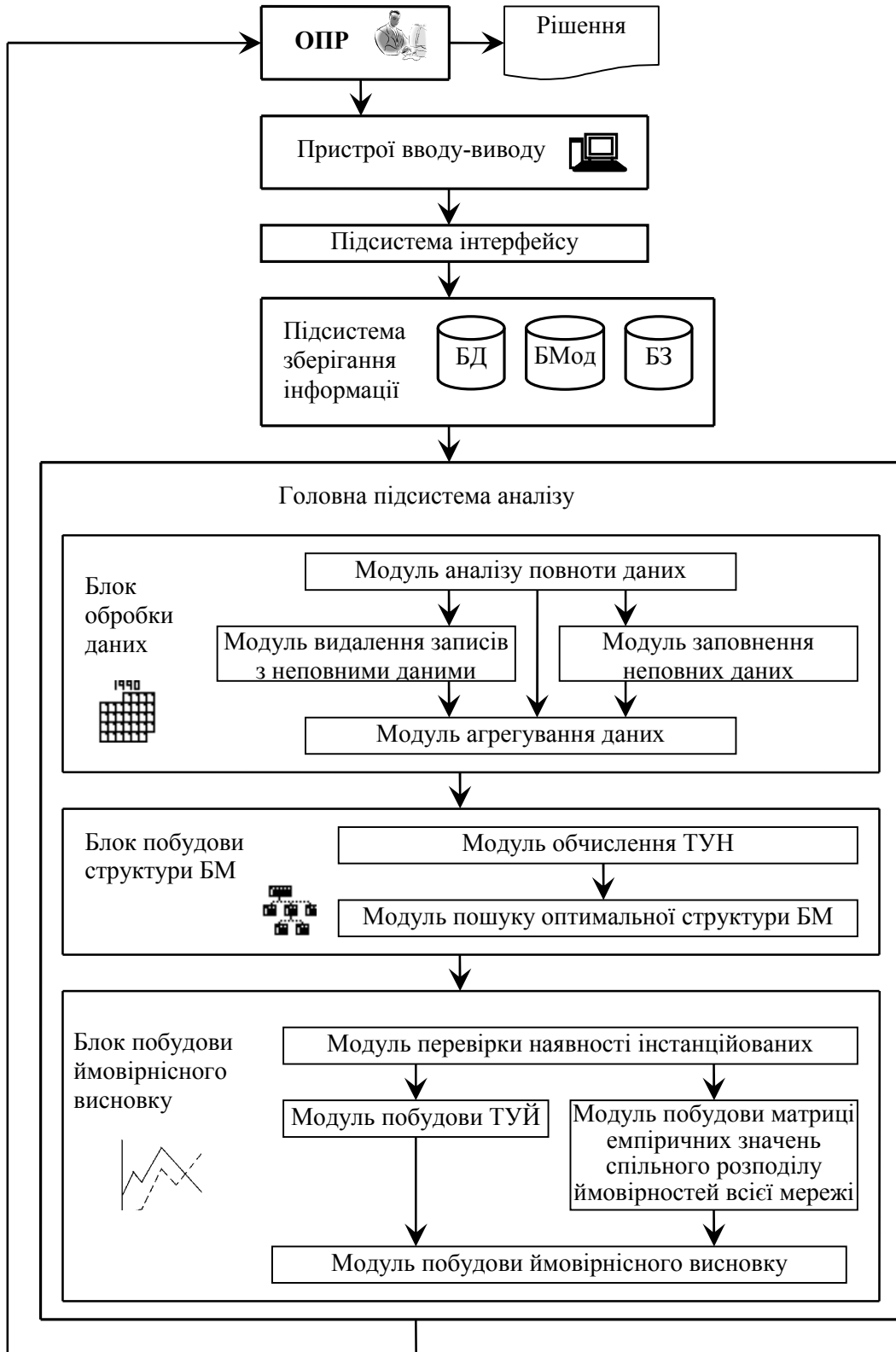


Рис. 6.7. Структура системи підтримки прийняття рішень для інтелектуального аналізу даних на основі мереж Байєса

Для всіх отриманих моделей досліджена низка ситуацій, для обчислення ймовірностей яких використано точний алгоритм побудови ймовірнісного висновку в мережі Байєса за навчальними даними.

За технічним рівнем дана система підтримки прийняття рішень для інтелектуального аналізу даних належить до *настільної СППР*, тобто ця система обслуговує лише один комп'ютер користувача [125], автором розробки не ставилася мета створення мережевого використання системи. За інструментальним підходом класифікації, розробленої Спрагом, дана система підтримки прийняття рішень є *СППР-генератором*, тому що кожна мережа Байєса, побудована на основі навчальних даних, представляє собою причинно-наслідкову мережу, що може розглядатись як окрема система підтримки прийняття рішень [126].

Архітектура системи є типовою для даного класу систем, вона складається з трьох основних підсистем і передбачає модульно-блочну побудову.

У рамках розробленої системи підтримки прийняття рішень для інтелектуального аналізу даних на основі мереж Байєса реалізовано спрощений варіант бази даних – текстова база даних, що являє собою сукупність навчальних даних у текстовому вигляді. Кожний такий файл включає всі навчальні дані для всіх змінних (атрибутів) процесу, тобто перший стовбець – дані для першої змінної, другий стовбець – дані для другої змінної і т. ін. Файл даних повинен містити не менше трьох стовпців даних, тобто процес описуватиметься не менш як трьома змінними. Дані в одному рядку (між стовпцями) слід відділяти один від одного пропуском, знаком табуляції або крапкою з комою. У дійсних числах для відокремлення дробової частини числа від цілої використовується кома, наприклад 10,2.

Змінні можуть містити не тільки кількісні (числові), але й якісні (рядкові) значення (наприклад, вербальні оцінки вигляду false/true або yes/no). Але в цьому випадку не допускаються пропуски усередині рядкового значення тому, що значення виду "Married more 10" при завантаженні даних призведе до некоректної роботи програми. У зв'язку з цим, рядкові значення, що складаються з декількох слів, повинні бути замінені на такі ж рядки значення, але без пропусків, знаків табуляції та крапок з комою. Наприклад, рядкове значення "Married more 10" можна замінити на "Married\_more\_10" (замість пропуску використовується нижній знак підкреслення) або "MarriedMore10" (кожне нове слово починається з великої букви). Редагувати довгі рядкові значення можна, наприклад, за допомогою програми Microsoft Excel або текстового редактора.

Комп'ютерна програма, що наведена в якості прикладу, реалізована із використанням мови програмування Delphi, що забезпечує підвищену надійність розроблюваного програмного продукту та зручність його подальшого супроводу.

Розроблена система підтримки прийняття рішень для інтелектуального аналізу даних на основі мереж Байєса використана для розв'язання задачі побудови скорингової моделі оцінювання **кредитоспроможності фізичних осіб при отриманні споживчого кредиту в одному з банків України**. Для побудови скорингової моделі використана реальна база даних клієнтів одного з українських комерційних банків. Усього в базі даних є 3347 записів про клієнтів. Як змінні процесу обрані десять атрибутів: стать; вік; сімейний стан; кількість дітей; чоловік (дружина) працює; освіта; тип зайнятості; поручитель; сума кредиту; результат



(кредитоспроможний?). На рис. 6.8 наведена система кредитного скорингу у вигляді мережі Байєса, побудованої за допомогою реалізованої програми.

Побудованій скоринговій моделі у вигляді мережі Байєса (рис. 6.7) відповідають наступні статистичні характеристики:

- 1 – похибка першого роду: 115;
- 2 – похибка другого роду: 157;
- 3 – загальна похибка: 272;
- 4 – загальна точність моделі: 0,918;
- 5 – похибка класифікації: 15%.

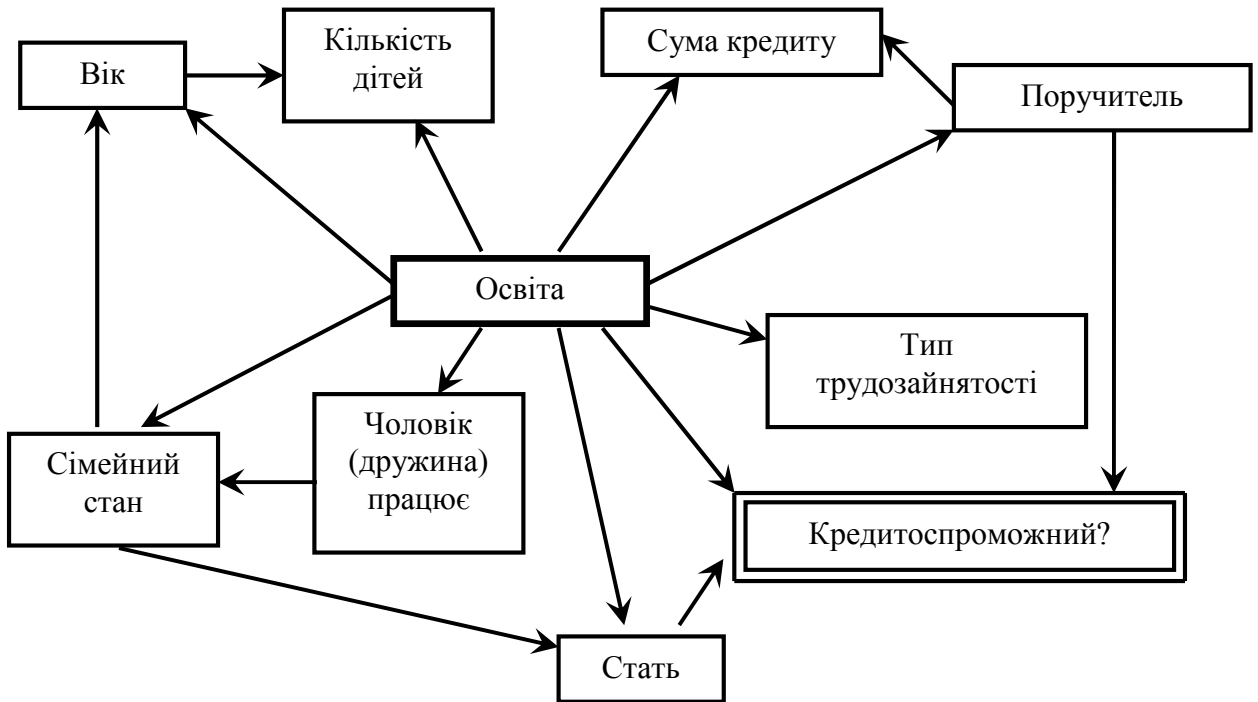


Рис. 6.8. Система кредитного скорингу у вигляді мережі Байєса

Тестова (перевірочна вибірка), що використовувалася для визначення якості прогнозу, складається із 100 записів. Відсоток похибок при класифікації дорівнює 15; це означає, що із 100 виданих кредитів, у 15-ти випадках позичальники були неправильно класифіковані.

Класифікація клієнтів за вибіркою, використаною для навчання, дала наступні результати:

- похибки 1-го роду дорівнюють 3,5% – прямі втрати банку, тобто, використовуючи дану модель, банк класифікував цих клієнтів як надійних, але вони виявилися некредитоспроможними;
- похибки 2-го роду дорівнюють 4,8 % – нереалізований дохід банку, тобто, використовуючи дану модель, банк класифікував цих клієнтів як ненадійних, але вони виявилися кредитоспроможними;
- загальна похибка складає 8,3%.

У таблиці 6.1 наведено десять ситуацій, для яких побудовано ймовірнісний висновок.

Таблиця 6.1

Результати моделювання по мережі Байєса, зображеній на рис. 6.7

Номер ситуації	Інстанційовані вершини (апріорна інформація, щодо клієнта)	Вірогідність того, що клієнт поверне кредит
1	Стать = "Чоловік"	92,08%
2	Стать = "Жінка"	97,55%
3	Поручитель = "Так"	99,06%
4	Поручитель = "Ні"	87,98%
5	Вік < 32 років та Сімейний стан = "Самотній" та Сума кредиту > 5000	76,92%
6	Тип трудовозайнятості = "Працівник банку" та Сімейний стан = "Одружений"	94,66%
7	Освіта = "Вища" та Кількість дітей = "один" та Чоловік (дружина) працює = "Так"	97,39%
8	Освіта = "Середня" та Кількість дітей = "немає" та Чоловік (дружина) працює = "Ні" та Поручитель = "Ні" та Сума кредиту > 2500	69,78%
9	Стать = "Чоловік" та Сімейний стан = "Удівець" та Освіта = "Середня спеціальна"	78,95%
10	Стать = "Жінка" та Сімейний стан = "Удівець" та Освіта = "Середня спеціальна"	98,81%

Для прогнозування композитного індексу нью-йоркської фондової біржі (<http://www.nyse.com>) сформована навчальна вибірка із 500 значень. На рис. 6.9 представлена побудована мережа Байєса.

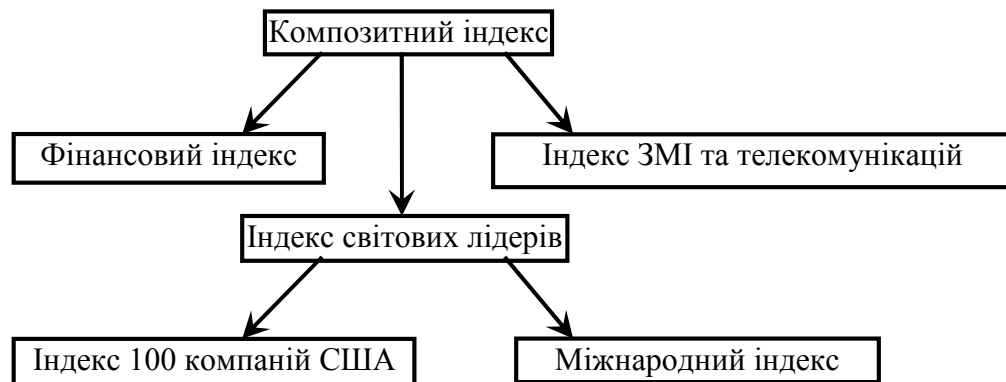


Рис. 6.9. Мережа Байєса для прогнозування композитного індексу нью-йоркської фондової біржі

В якості факторів аналізу процесу розглядалися наступні індекси: композитний, індекс 100 компаній США, індекс енергетики, індекс 100 світових компаній, фінансовий індекс, індекс світових лідерів, індекс охорони здоров'я, індекс засобів масової інформації та телекомунікацій. Композитний індекс відображає зміни курсів усіх акцій, що представлені на Нью-Йоркській фондовій біржі, які включають більше 1500 найбільших американських компаній із загальною капіталізацією більше 20 трильйонів доларів. Цей індекс є зручним показником стану економіки США. Як видно з рис. 6.8, не всі індекси беруть участь у

прогнозуванні композитного індексу. У процесі роботи розробленої системи підтримки прийняття рішень було відібрано найбільш значущі індекси з множини навчальних даних, за якими побудована мережа Байєса.

На рис. 6.10 наведена побудована мережа Байєса із значеннями вершин, отриманими за вибіркою навчальних даних.

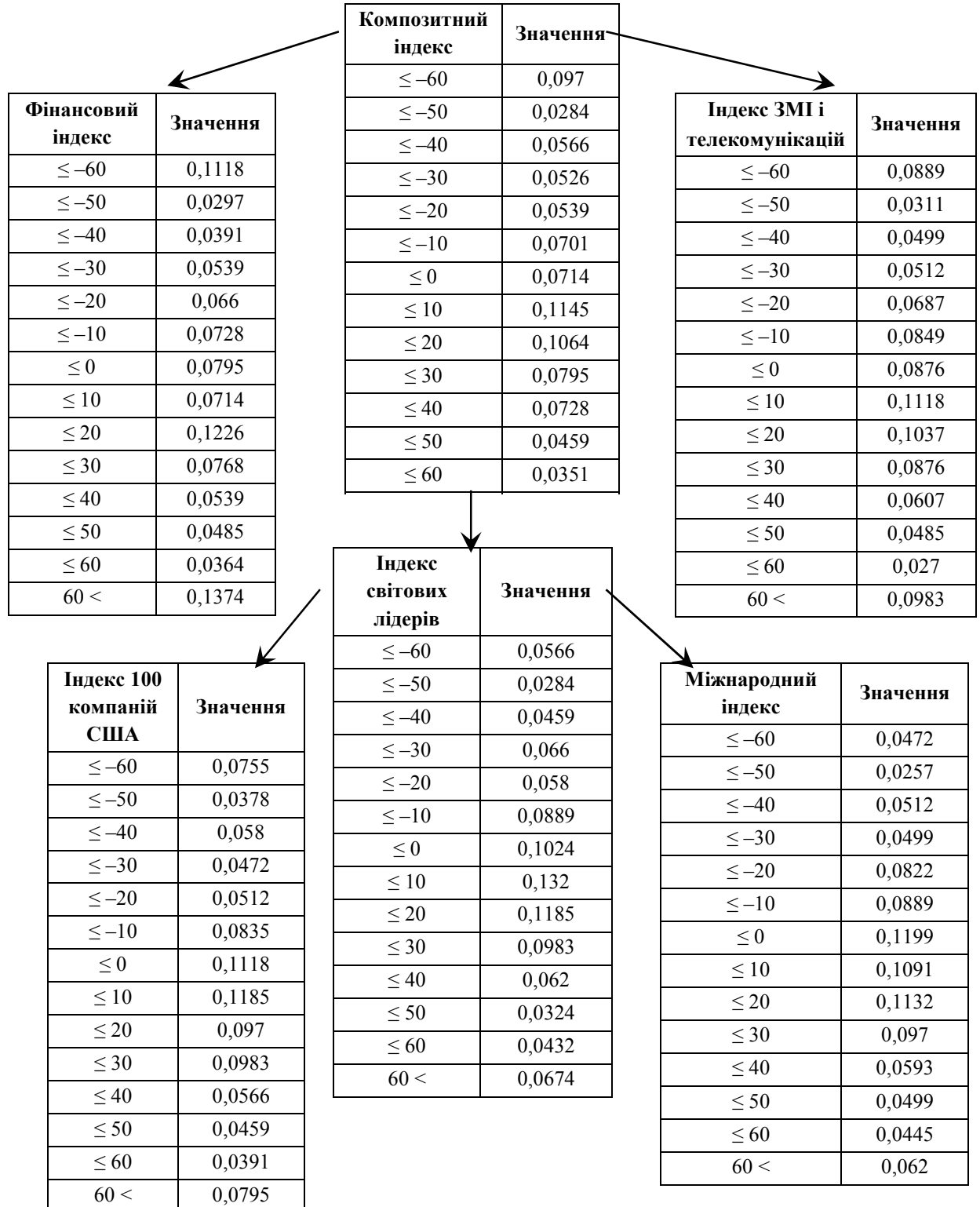


Рис. 6.10. Мережа Байєса із значеннями ймовірностей вершин

Фінансовий індекс показує зміни курсу непривілейованих акцій фінансового сектору США та сільськогосподарських підприємств. Індекс засобів масової інформації та телекомунікацій показує зміни 100 компаній лідерів із сектора ЗМІ та телекомунікацій. Представлені в цьому індексі компанії мають капіталізацію ринку в розмірі 2,3 трильйонів доларів.

Індекс світових лідерів відображає зміни курсу непривілейованих акцій 200 найбільших компаній світу, включаючи 100 найбільших компаній США, із загальною капіталізацією 9,7 трильйонів доларів з 10 промислових секторів всіх регіонів світу (за станом на травень 2008 року в цьому індексі представлено 19 країн). Індекс відображає диверсифікованість акціонерного капіталу і допомагає слідкувати за інвестиційними ринками світу.

Індекс топ-100 компаній США показує зміни курсу непривілейованих акцій 100 найбільших компаній США із загальною капіталізацією 6,7 трильйонів доларів.

Міжнародний індекс відображає зміни курсу непривілейованих акцій 100 найбільших не американських компаній світу із загальною капіталізацією 5,1 трильйонів доларів.

Для отриманої моделі прогнозування композитного індексу у вигляді мережі Байєса (рис. 6.8 та 6.9) було виконано 100 ітерацій прогнозування. На рис. 6.11 наведена структурна схема побудови прогнозу. У 96% випадків точно спрогнозовано зростання або спад значення композитного індексу, а в 52% випадків прогнозне та реальне значення відхилення композитного індексу від значення в попередній момент часу повністю збіглися.

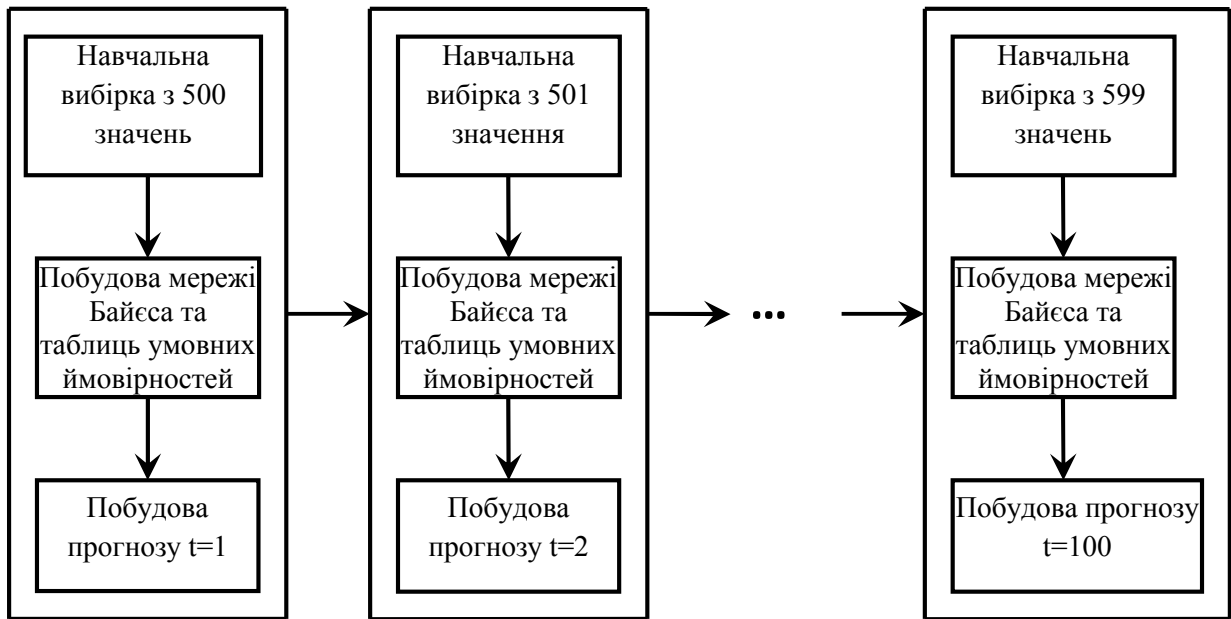


Рис. 6.11. Структурна схема прогнозування композитного індексу мережами Байєса на 100 часових інтервалах

На діаграмі 6.12 представлені зміни прогнозного та реального відхилень композитного індексу на перших 50 часових інтервалах.



Рис. 6.12. Діаграма зміни прогнозного та реального відхилень композитного індексу на 50 часових інтервалах

### Контрольні питання

1. Нарисуйте структуру системи підтримки прийняття рішень на основі мереж Байєса
2. Чи можна побудувати інтелектуальну систему підтримки прийняття рішень на основі байєсівських мереж? Відповідь обґрунтуйте.
3. Які задачі можна розв'язувати за допомогою систем підтримки прийняття рішень, побудованої з використанням байєсівських мереж?
4. Які критерії застосовують при аналізі якості структури та ймовірнісного висновку?
5. Поясніть суть байєсівського інформаційного критерію якості.
6. Нарисуйте структуру системи підтримки прийняття рішень на основі мереж Байєса.
7. Охарактеризуйте узагальнену концептуальну модель системи підтримки прийняття рішень.
8. Назвіть основні компоненти системи підтримки прийняття рішень та їх особливості за використання байєсівських мереж
9. Наведіть приклади практичної реалізації байєсівських мереж в банківській сфері, вкажіть на їх особливості.
10. Опишіть медичну експертну систему у вигляді мережі Байєса.
11. Назвіть системи класифікації систем підтримки прийняття рішень. До якої групи належать СППР на основі мереж Байєса
12. Для чого призначений модуль пошуку оптимальної структури мережі Байєса.
13. Назвіть особливості систем підтримки прийняття рішень комбінованого типу.
14. Охарактеризуйте основні підходи щодо використання обчислювальних алгоритмів у системах підтримки прийняття рішень.
15. Виконання яких вимог передбачає системний підхід до проектування інформаційних систем підтримки прийняття рішень.

## РОЗДІЛ 7 МЕРЕЖІ БАЙЄСА ЯК ІНСТРУМЕНТ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ОБРАЗІВ

- Задача розпізнавання образів
- Методи розпізнавання образів
- Класифікація методів розпізнавання образів
- Застосування систем розпізнавання образів
- Розпізнавання образів на основі наївного байєсівського класифікатора
- Використання динамічної мережа Баєйса для розпізнавання образів

### 7.1. Задача розпізнавання образів

Історично склалося так, що більшість задач, пов'язаних із розпізнаванням, зокрема метеопадів, авіаційної ситуації в районі аеропорту авіадиспетчером, смуги посадки льотчиком у складних метеорологічних умовах тощо, як правило, людина вирішує ефективно, тобто з прийнятною якістю. Слід зауважити, що в процесі зазначеної діяльності людини кількість прийнятих рішень за результатами розпізнавання ситуацій є скінченною. Тоді як множина станів зовнішнього середовища, оцінюваних у процесі безпосередньо розпізнавання, що призводять до зазначених розв'язків, може бути нескінченною. Це можна бачити на прикладі друкарки, яка друкує під диктування. З незліченної множини варіантів вимови одного й того ж звуку вона вибирає тільки один, завжди вдаючись до клавіші друкарської машинки, на якій зображено літеру, що позначає даний звук. У результаті вона безпомилково друкує слова, незалежно від їхнього спотворення при усному мовленні.

До сприйняття такої скінченної кількості розв'язків людина підготовлена усім своїм життєвим досвідом. Тому сприйняття ідеології автоматизації процесів, пов'язаних із вирішенням задач розпізнавання, розуміння можливості та доцільності усунення людини як ланки з автоматичних систем, призвело до того, що людство насамперед зосередилось на створенні автоматів, здатних реагувати на множину змін характеристик зовнішнього середовища деякою обмеженою кількістю раціональних розв'язків (реакцій) виконавчих органів цих автоматів.

У результаті були одержані пристрої, що розпізнають різні об'єкти, можуть підвищувати, наприклад, ефективність системних зв'язків (розпізнаючи сигнали в шумах), допомагають встановлювати об'єктивний діагноз захворювань, дають можливість здійснювати автоматичний контроль складних технічних систем, вчасно втручатися в їх роботу та виконувати ремонтно-відновлювальні роботи і т. ін.

Основні цілі заміни людини в задачах розпізнавання образів зводяться до таких:

- звільнення людини від одноманітних рутинних операцій для розв'язання інших більш важливих задач;
- підвищення якості виконуваних робіт;
- підвищення швидкості розв'язання задач.

У задачах розпізнавання символів сьогодні досить часто застосовуються методики, основані на таких моделях, як наївний байєсівський класифікатор,

мережа Байєса зі зворотним формуванням імовірнісного висновку та динамічна мережа Байєса, а саме – випадок прихованої моделі Маркова.

**Розпізнавання образів** можна визначити як віднесення вихідних даних до певного класу за допомогою виділення суттєвих ознак або властивостей, що характеризують ці дані, із загальної маси несуттєвих подробиць [7].

Ці задачі є інформаційними і складаються із двох етапів:

- 1) перетворення вихідних даних до вигляду, зручного для розпізнавання;
- 2) власне розпізнавання належності об'єкта певному класу.

Зокрема прогноз погоди можна інтерпретувати як задачу розпізнавання образів. Вихідні дані в цьому випадку приймають вид синоптичних карт. Система інтерпретує їх, виділяючи суттєві ознаки і формуючи на їх основі прогноз.

Постановку медичного діагнозу також можна розглядати як задачу розпізнавання образів. Симптоми слугують вихідними даними для системи, що розпізнає, яка на основі їх аналізу ідентифікує захворювання.

Система розпізнавання символів представляє собою систему розпізнавання образів, у яку в якості вихідних даних вводяться оптичні сигнали і яка ідентифікує їх назви. У системі розпізнавання мови сказане слово ідентифікується за допомогою аналізу сприйнятого системою звукового сигналу.

**Розпізнаванням образів** називають задачі побудови і застосування формальних операцій над числовими або символічними зображеннями об'єктів реального або ідеального світу, результати розв'язання яких відображають відносини еквівалентності між цими об'єктами. Відносини еквівалентності виражають належність оцінюваних об'єктів до яких-небудь класів, що розглядаються як самостійні семантичні одиниці [7].

При побудові алгоритмів розпізнавання класи еквівалентності можуть задаватися дослідником, який користується власними змістовними уявленнями або використовує зовнішню додаткову інформацію про подібність і відмінність об'єктів у контексті розв'язуваного завдання. Тоді говорять про "розпізнавання з учителем" [177]. А якщо ні, то, тобто коли автоматизована система вирішує завдання класифікації без залучення зовнішньої навчальної інформації, говорять про автоматичну класифікацію або "розпізнавання без учителя". Процес розпізнавання багатокроковий та ітераційний, потребує значних обчислювальних потужностей.

**Образ** – це опис будь-якого елемента як представника відповідного класу образів [7, 177]. У випадку, коли множина образів розділяється на класи, які не перетинаються, бажано використовувати для віднесення цих образів до відповідних до класів який-небудь автоматичний пристрій.

Прикладом такої задачі є зчитування й обробка погашених банківських чеків. Подібні задачі можуть виконуватися й людьми, але машина справляється з ними набагато швидше. З іншого боку, деякі задачі розпізнавання є такими, що людина навряд чи у змозі їх вирішувати. Прикладом задач такого роду є виділення з множини морських сигналів і шумів тону підводного човна шляхом аналізу підводних звукових сигналів. Очевидне, але зовсім уже "нехитре" розв'язання задачі розпізнавання полягає в застосуванні до окремих пред'явлених образів ряду простих тестів для виділення ознак кожного класу. Сукупність цих тестів повинна розрізняти всі припустимі образи з різних класів. Задачі розпізнавання мають такі характерні риси:

- до задач даного класу можна вводити поняття аналогії або подібності об'єктів і формулювати правила, на підставі яких об'єкт зараховується до одного і того ж самого класу або до різних класів;

- у цих задачах можна оперувати набором прецедентів – прикладів, класифікація яких уже відома, і які у вигляді формалізованих описів можуть бути пред'явлені алгоритму розпізнавання для налаштування на задачу в процесі навчання;

- для цих задач важко будувати формальні теорії й застосовувати класичні математичні методи (часто недоступна інформація для точної побудови математичної моделі або виграш від використання моделі і математичних методів незначний порівняно з витратами).

## 7.2. Методи розпізнавання образів

В основі ідеї синтезу систем автоматичного розпізнавання лежать способи, за допомогою яких описують і розподіляють класи образів. Коли клас характеризується переліком вхідних членів, побудова системи розпізнавання образів може ґрунтуватися на принципі належності до цього переліку. Коли клас характеризується деякими загальними властивостями, притаманними всім його членам, побудова системи розпізнавання може ґрунтуватися на принципі спільності властивостей. Якщо при розгляді класу виявляється тенденція до утворення кластерів у просторі образів, побудова системи розпізнавання може ґрунтуватися на принципі кластеризації.

Під кластером розуміють зазвичай групу об'єктів (образів), що утворюють у просторі опису компакту у деякому значенні область. Ці **три основні принципи побудови систем розпізнавання образів** є наступними.

1. *Принцип послідовного порівняння з відомими членами класу.* Задавання класу образів для порівняння припускає реалізацію процесу автоматичного розпізнавання образів шляхом зрівнювання з еталоном. Множина образів, що належать одному класу, запам'ятовується системою розпізнавання. При пред'явленні системі незнайомих (нових) образів вона послідовно порівнює їх з образами, що зберігаються в її пам'яті. Так, наприклад, якщо в пам'ять системи розпізнавання введено літери різних складальних шрифтів, то підхід, заснований на перерахуванні членів класу, дозволяє розпізнавати відповідні літери, але тільки в тих випадках, коли їх зображення не перекручені шумом, пов'язаним з розмазуванням або поганим нанесенням фарби, пористістю паперу і т. ін. Безсумнівно, це нескладний метод, однак він дозволяє будувати недорогі системи розпізнавання, які в окремих прикладних областях цілком справляються із своїми завданнями. Метод послідовного порівняння з членами класу працює задовільно, якщо вибірка образів близька до ідеальної.

2. *Принцип спільності властивостей.* Задавання класу за допомогою властивостей, загальних для всіх його членів, передбачає реалізацію процесу автоматичного розпізнавання шляхом виділення подібних ознак і роботи з ними. Основне допущення в цьому методі полягає в тому, що образи, які належать одному і тому ж класу, мають ряд спільних властивостей або ознак, що відображають подібність таких образів. Ці спільні властивості можна ввести зокрема в пам'ять системи розпізнавання. Коли системі пред'являється некласифікований образ, то



виділяється набір ознак, що його описують, при цьому останні іноді кодуються, а потім вони порівнюються з ознаками, закладеними в пам'ять системи розпізнавання. У такому випадку остання зарахує пред'явлений для розпізнавання образ у клас, що характеризується системою ознак, подібних до ознак цього образу. Отже, при використанні даного методу основне завдання полягає у виділенні ряду загальних властивостей за скінченою вибіркою образів, належність яких до шуканого класу відома.

Очевидно, що ця концепція розпізнавання у багатьох відношеннях перевершує розпізнавання за принципом послідовного порівняння з членами класу. Для запам'ятовування ознак класу потрібно значно менше пам'яті, ніж для зберігання всіх об'єктів, що входять у клас. Оскільки ознаки, що характеризують клас у цілому, мають інваріантність, принцип порівняння ознак допускає варіацію характеристик окремих образів. Процедура порівняння з еталоном, з іншого боку, не допускає істотних варіацій характеристик окремих образів. Якщо всі ознаки, що визначають клас, можна знайти за наявною вибіркою образів, то процес розпізнавання зводиться просто до порівняння з ознаками. Задача стає винятково важкою, навіть не невирішуваною взагалі, оскільки складно знайти для деякого класу повний набір ознак для розпізнавання. Отже, застосування цього принципу розпізнавання часто пов'язане з необхідністю розвитку методів вибору ознак, що є в деякому сенсі оптимальними.

3. Принцип кластеризації. Коли образи деякого класу представляють собою вектори, компонентами яких є дійсні числа, цей клас можна розглядати як кластер і виділяти тільки його властивості у просторі образів кластера. Побудова систем розпізнавання, заснованих на реалізації даного принципу, визначається взаємним просторовим розташуванням окремих кластерів. Якщо кластери, що відповідають різним класам, рознесені досить далеко один від одного, то можна з успіхом скористатись порівняно простими схемами розпізнавання, наприклад, такими як класифікація за принципом мінімальної відстані. Якщо ж кластери перекриваються, то доводиться звертатись до складніших методів розбивки простору образів. Перекриття кластерів є результатом неповноцінності доступної інформації й шумових викривлень результатів виміру. Тому ступінь перекриття часто вдається зменшити шляхом збільшення кількості і якості вимірів, що виконуються для образів деякого класу.

### 7.3. Класифікація методів розпізнавання образів

Існує багато типологій методів розпізнавання образів. Зокрема, розрізняють параметричні, непараметричні та евристичні методи. За іншою типологією виділяють групи методів, виходячи з історично створених шкіл і напрямів у даній області. Наприклад, у роботі [178], дано широкий огляд методів розпізнавання образів та запропоновано їх типологію:

- методи, засновані на принципі розділення;
- статистичні методи;
- методи на основі "потенційних функцій";
- методи обчислення оцінок (голосування);
- методи, засновані на обчисленні висловлювань, зокрема на апараті алгебри логіки.

Д. А. Поспелов [179] виділяє два основні *способи представлення знань*.

1. Іntenціональне представлення – у вигляді схеми зв'язків між атрибутами (ознаками).

2. Екстенціональне представлення – за допомогою конкретних фактів (об'єкти, приклади).

Іntenціональне представлення фіксує закономірності й зв'язки, якими пояснюється структура даних. Стосовно діагностичних задач така фіксація полягає у визначенні операцій над атрибутами (ознаками) об'єктів, що призводить до необхідного діагностичного результату. Іntenціональні представлення реалізуються за допомогою операцій над значеннями атрибутів і не передбачають виконання операцій над конкретними інформаційними фактами (об'єктами). У свою чергу, екстенціональні представлення знань пов'язані з описом і фіксацією конкретних об'єктів із предметної області й реалізуються в операціях, елементами яких служать об'єкти як цілісні системи.

Описані вище два фундаментальних способи представлення знань дають можливість запропонувати таку класифікацію методів розпізнавання образів.

1. Іntenціональні методи розпізнавання образів – методи, засновані на операціях з ознаками.

2. Екстенціональні методи розпізнавання образів – методи, засновані на операціях з об'єктами.

До першої групи методів належать:

– засновані на оцінках густини розподілу значень ознак (або подібності й відмінності об'єктів);

– засновані на припущеннях стосовно класу вирішальних функцій;

– логічні методи;

– структурні (лінгвістичні) методи.

До другої групи:

– порівняння з прототипом;

–  $k$ -найближчих сусідів;

– алгоритми обчислення оцінок (голосування);

– колективи вирішальних правил.

Інша класифікація ґрунтується на реалізації розглянутих вище основних принципів побудови автоматичних систем розпізнавання образів. Розрізняють три основні типи методології: евристична, математична і лінгвістична (синтаксична). Нерідко системи розпізнавання створюються на основі комбінації цих методів.

### ***Евристичні методи***

За основу евристичного підходу взяті інтуїція й досвід людини; у ньому використовуються принципи послідовного порівняння членів класу й спільності властивостей. Зазвичай системи, побудовані за такими методами, включають множину специфічних процедур, розроблених відповідно до конкретних задач розпізнавання. Прикладом подібного підходу є розв'язання задачі розпізнавання символів, що передбачає класифікацію образу (символу) на основі виділення окремих ознак, зокрема таких, як кількість і послідовність розташування специфічних штрихів (рисок). Хоча евристичний підхід відіграє більшу роль у побудові систем розпізнавання образів, небагато можна сказати щодо загальних принципів синтезу, оскільки розв'язок кожної конкретної задачі вимагає

використання специфічних прийомів розробки системи. Таким чином, це означає, що структура і якість евристичної системи у значній мірі визначаються знаннями і досвідом розробників.

### *Математичні методи*

В основу математичного підходу покладено правила класифікації, які формулюються й виводяться в рамках певного математичного формалізму за допомогою принципів спільності властивостей і кластеризації. Цим даний підхід відрізняється від евристичного, у якому розв'язки визначаються за допомогою правил, щільно пов'язаних з характером розв'язуваної задачі. Математичні методи побудови систем розпізнавання можна розділити на два класи: детерміністські й статистичні.

Детерміністський підхід ґрунтується на математичному апараті, що не використовує у явному вигляді статистичні властивості досліджуваних класів образів.

Статистичний підхід ґрунтується на математичних правилах класифікації, які формулюються і синтезуються в термінах математичної статистики. Побудова статистичного класифікатора в загальному випадку припускає використання байєсівського класифікаційного правила і його різновидів. Це правило забезпечує створення оптимального класифікатора у тих випадках, коли відомі щільності розподілу для всіх сукупностей образів і ймовірності появи образів для кожного класу.

У даному випадку постановка задачі розпізнавання образів формулюється наступним чином. Нехай задана деяка множина образів, кожний з яких належить одному з  $w$  класів. Спосіб розподілення на класи достовірно невідомий, але задані кінцеві вибірки образів, для яких заздалегідь відома належність до деякого класу. Задача полягає в тому, щоб класифікувати будь-який новий образ. Для цього створюють належне правило прийняття рішення. Основою для такого вибору є теорія статистичних рішень. Істинні імовірності характеристик класів зазвичай невідомі. Тому на практиці використовують оцінки, отримані за відомими прикладами класів. Цю звичайну статистичну процедуру оцінювання за вибіркою імовірнісних характеристик генеральної сукупності тут інтерпретують як навчання. Вирішуючі правила фактично не залежать від специфіки об'єктів, що розпізнаються, оскільки вони мають справу не з самими об'єктами, а з деякими числовими характеристиками. Внаслідок цього ці правила мають відому узагальненість.

Методи статистичних рішень призводять до вирішуючих процедур, оптимально задовольняючи наперед заданому критерію, і найкращим чином використовують відомі параметри. Критерієм оптимальності висувається вимога, щоб загальна імовірність неправильного розпізнавання була мінімальною. Цей критерій рівносильний критерію Байєса, якщо апостеріорні імовірності класів рівні між собою; ціни помилкових класифікацій рівні між собою, а результуюча межа рішень є контуром, на якому функції густини для пари класів мають одне й те ж саме значення.

В якості одного з можливих критеріїв статистичної перевірки гіпотез можна прийняти критерій Байєса. Саме він був обраний Є. І. Соколовим [180] у його імовірнісній моделі сприйняття. Серед можливих оптимальних вирішальних функцій найбільш докладно розглянуто її лінійні наближення, тобто лінійні

вирішальні функції. Лінійність тут означає, що пари узагальнених образів дихотомічно розділяються в просторі образів однієї й тільки однієї гіперплощини вигляду:

$$\sum_{i=1}^n \lambda_i x_i - \lambda_{i+1} = 0, \tag{7.1}$$

де  $x_i$  – координати простору образів.

До переваг таких вирішальних функцій необхідно віднести те, що їх можна емпірично конструювати, не роблячи ніяких припущень щодо вимірів, проведених при сприйнятті образу, що відповідно орієнтується, а також стосовно апріорних ймовірностей появи класів образів за умови, що є в наявності джерело образів. Крім того, як уже відмічалось, реалізація лінійних вирішальних функцій дуже проста, і для технічної системи розпізнавання не потрібно великих обчислювальних витрат.

Використання лінійних вирішальних функцій добре зарекомендувало себе при розв'язанні багатьох задач розпізнавання. У загальному вигляді задача було розглянуто Ю. В. Вагіним [181]. Тут алгоритм розпізнавання будувався як метод послідовного багатоальтернативного аналізу. За основу процесу розпізнавання образів прийнято перерозподіл вихідних взаємовиключних ймовірностей  $\{p_i\} = p_1, p_2, p_3, \dots, p_w$  таких, що представляють повну групу класів  $\{S_i\} = S_1, S_2, S_3, \dots, S_w$ , до яких може бути віднесений розпізнаваний образ, тобто:

$$\left. \begin{aligned} 0 \leq p_i \leq 1 \\ \sum_{i=1}^w p_i = 1 \end{aligned} \right\}, \tag{7.2}$$

і, крім того,

$$\{p(s_x \in S_i)\} = p(s_x \in S_1), p(s_x \in S_2), \dots, p(s_x \in S_w), \tag{7.3}$$

для яких виконується умова нормування:

$$\sum_{i=1}^w p(s_x \in S_i) = 1. \tag{7.4}$$

Належність  $s_x$  до одного з класів встановлюється за умови, що множина ймовірностей приналежності  $s_x$  до класів  $S_1, S_2, S_3, \dots, S_w$  задовольняє вимогам, спричиненим необхідною достовірністю відповідей і полягають, наприклад, у тому, що для деякого значення  $i$  виконуються всі нерівності:

$$\frac{p(s_x \in S_i)}{p(s_x \in S_j)} \geq A_{ij}, \tag{7.5}$$

$$j=1, 2, 3, \dots, w,$$

де  $A_{ij}$  – деяке додатне число.

Інакше кажучи, в даному методі за основу процесу розпізнавання беруть процес перерозподілу вихідних ймовірностей за допомогою аналізу сукупності отриманих реалізацій випадкового процесу, що представляє собою образ, який намагаються розпізнати, і отримання за рахунок цього необхідної інформації.

Необхідно окремо зазначити, що всі опубліковані статистичні моделі розпізнавання образів, що використовують фактично різні статистики (відношення

подібності, дискримінантні функції та деякі інші), дають дуже хороші результати розпізнавання в різних задачах – медичної діагностики, розпізнаванні рукописних символів, класифікації даних сейсмозвідки і т. ін.

### Структурні (лінгвістичні) методи

Ці методи ґрунтуються на використанні спеціальних граматики породжуючих мов, за допомогою яких можна описувати сукупність властивостей об'єктів, що розпізнаються. Для різних класів об'єктів виділяються непохідні (атомарні) елементи (підобрази, ознаки) і можливі взаємозв'язки між ними. Граматикою називають правила побудови об'єктів з цих непохідних елементів. Таким чином, кожен об'єкт представляється сукупністю непохідних елементів, "сполучених" між собою тими або іншими способами, або, іншими словами – "реченням" деякої "мови". При цьому важливою є інформація, яка описує структуру кожного об'єкта, а до процедури розпізнавання ставиться вимога щоб вона надавала можливість не лише віднести об'єкт до певного класу, але й описати ті властивості об'єкта, які унеможливають віднесення його до іншого класу. Типовим прикладом є розпізнавання зображень. Об'єкти, які розглядаються в цьому класі задач, є складними і кількість необхідних ознак є великим. Це робить привабливою ідею описання складного об'єкта у вигляді ієрархічної структури простіших підобразів. Наприклад, зображення на рис. 7.1 може бути описано за допомогою ієрархічних структур, показаних на рис. 7.2.

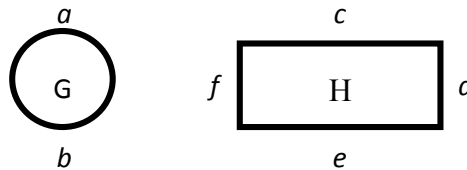


Рис. 7.1. Зображення  $F$

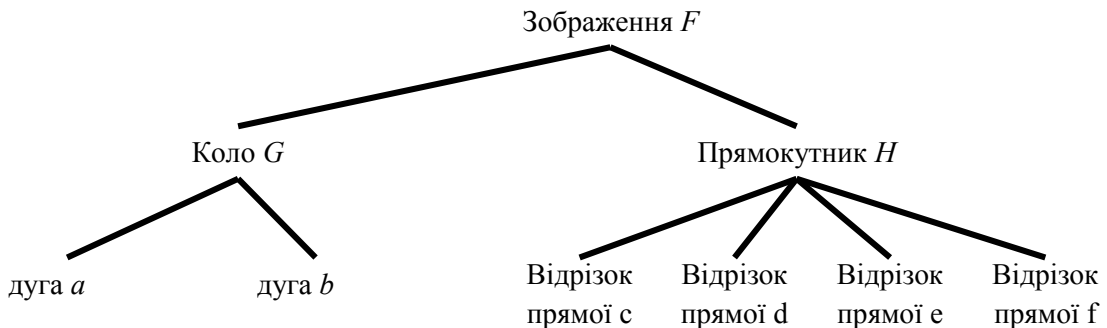


Рис 7.2. Ієрархічний структурний опис зображення  $F$

Процес розпізнавання здійснюється після ідентифікації у об'єкті непохідних елементів і складанні опису об'єкта. Розпізнавання полягає у синтаксичному аналізі або у граматичному розборі "речення", що описує даний об'єкт. Ця процедура встановлює, чи є це речення синтаксично правильним по відношенню до заданої граматики. Паралельно синтаксичний аналіз дає деякий структурний опис речення (зазвичай у вигляді деревовидної структури).

Різні відношення, визначені між підобразами, або операції композиції, звичайно можуть бути виражені логічними та (або) математичними операціями.

Прикладом такого відношення може бути операція конкатенації. Речення може бути представлене за допомогою деревовидної структури або графа відношень. На рис. 7.3 показано граф відношень для зображення  $F$  (рис. 7.1). Використання графа відношень дає можливість розширити клас допустимих відношень та включити в нього будь-яке відношення, яке зручно визначається з образу. Однак використання деревовидних структур дозволяє безпосередньо пристосувати методи теорії формальних мов до розв'язання задачі компактного представлення і аналізу образів, які мають істотний структурний зміст.

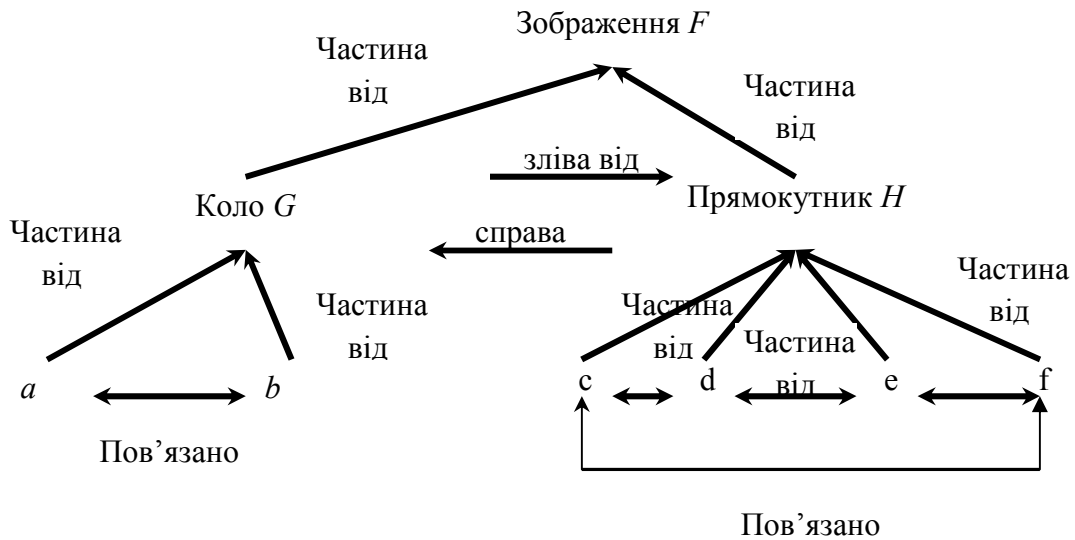


Рис 7.3. Граф відношень зображення  $F$

Систему синтаксичного розпізнавання образів можна вважати такою, що складається з трьох основних частин, а саме: з блоку попередньої обробки, блоку опису або представлення об'єкта і блоку синтаксичного аналізу. Блок попередньої обробки здійснює, по-перше, функції кодування й апроксимації і, по-друге, функції фільтрації, відновлення та поліпшення стану об'єкта. Кожний підданий попередній обробці об'єкт потім представляють у вигляді мовної структури. Цей процес представлення об'єкта включає, по-перше, сегментацію, а по-друге, – виділення непохідних елементів (ознак). Для того, щоб отримати граматику, що описує структурну інформацію класу образів, необхідно мати пристрій виводу граматики, що дозволяє відновити її по заданій множині об'єктів у вигляді, подібному мовному. Функції цього пристрою аналогічні процесу "навчання" у дискримінантних системах розпізнавання образів. У результаті навчання на прикладах об'єктів розглянутого класу формується структурний опис цього класу. Потім отриманий у вигляді граматики опис використовується для опису образів і синтаксичного аналізу. Очевидно, що для знаходження найкращої множини непохідних елементів і відповідного структурного опису класу розглянутих образів, необхідно виконати навчання більш загального виду.

#### 7.4 Застосування систем розпізнавання образів

У табл. 7.1 перераховано ряд задач класифікації, а також відповідні вихідні дані й відповіді, які видає система.

Таблиця 7.1

## Класифікація задач розпізнавання

Характер задачі розпізнавання	Вид вихідних даних	Вид відповіді системи розпізнавання
Розпізнавання символів	Оптичні сигнали	Назви символів
Розпізнавання мови	Акустичні сигнали, голос	Слова людини, яка говорить
Прогноз погоди	Синоптичні карти	Прогноз погоди
Встановлення медичного діагнозу	Симптоми захворювання	Вид захворювання

Сьогодні важко назвати таку галузь науки і сфери виробництва, де не використовуються або не будуть використовуватись системи розпізнавання. Розглянемо деякі відомі застосування.

Системи технічної діагностики. Їх впровадження у практику – найважливіший фактор підвищення ефективності використання машин і технологічного обладнання, різкого скорочення витрат на експлуатацію. Тенденція ускладнення, а значить і подорожчання машин, що виникло історично, постійно збільшує витрати на експлуатацію. Вихід з цієї ситуації полягає в переході до систем технічної діагностики (розпізнавання стану машин), наприклад, безрозбірного пошуку неполадок та відмов. У результаті застосування такого підходу замість планово-запобіжного ремонту виконується ремонт за фактичною необхідністю. Наприклад, в інструкції для експлуатації автомобіля передбачені планові технічні обслуговування через 500 км, 1000 км, 2000 км і т. ін. Якщо ж його оснастити системами розпізнавання станів, то від планових технічних оглядів можна було б відмовитися, замінивши їх обслуговуванням окремих вузлів і систем за необхідністю.

Автоматизовані системи діагностики в медицині – це шлях збільшення:

- широти й глибини охоплення симптомів (розраховувати тільки на пам'ять лікаря у всіх ситуаціях дуже важко; краще функцію пам'яті віддати комп'ютеру);
- оперативності (комп'ютер забезпечить майже миттєвий результат);
- достовірності (діагноз комп'ютера не залежить від зовнішніх факторів, як це трапляється з людиною).

У сільському господарстві області застосування такі:

- розпізнавання об'ємів урожаю за даними космічних спостережень;
- зменшення ручної праці при сортуванні плодів за формою, кольору й розмірами і т. ін.

На сьогодні вже теоретично розроблено низку різних методів розпізнавання образів. Однак для більшості цих методів їх програмна реалізація відсутня. Реально на ринку можна реалізувати тільки такі програмні системи, в основі яких лежать досить прості й "прозорі" математичні моделі. Тому розробник, зацікавлений у тиражуванні свого програмного продукту, приходять до задачі вибору математичної

моделі не із чисто наукової точки зору, а як прагматик із урахуванням можливостей програмної реалізації.

Далі в розділі описуються методики для розпізнавання символів на основі таких моделей, як наївний байєсівський класифікатор, мережа Байєса зі зворотним формуванням імовірнісного висновку та динамічна мережа Байєса, а саме випадок прихованої моделі Маркова.

Перші замітки про приховані моделі Маркова опублікував Баум в 1960-х, і вже в 70-х їх уперше застосували при розпізнаванні мови. З середини 1980-х ПММ застосовуються для аналізу біологічних послідовностей, зокрема ДНК. Крім того, приховані моделі Маркова застосовуються в крипто-аналізі та машинному перекладі.

### 7.5. Системи розпізнавання образів на основі наївного байєсівського класифікатора

Наївний байєсівський класифікатор – це простий імовірнісний класифікатор, який ґрунтується на застосуванні теореми Байєса із строгим (наївним) припущенням про незалежність.

Залежно від точної природи імовірнісної моделі, наївні байєсівські класифікатори можуть навчатися досить ефективно. Незважаючи на значно спрощені умови, наївні байєсові класифікатори часто працюють набагато краще у багатьох складних життєвих ситуаціях.

Перевагами наївного байєсового класифікатора є невеликий об'єм навчальних даних, які потрібні для оцінки параметрів, необхідних для класифікації.

#### Модель наївного байєсівського класифікатора

Нехай змінна може приймати значення з деякої множини значень  $V$ . Змінна  $x$  описується змінними  $\langle a_1, a_2, \dots, a_n \rangle$  – атрибутами. Необхідно знайти найбільш імовірне значення  $x$ , тобто:

$$v_0 = \arg \max_{v \in V} P(x = v | a_1, a_2, \dots, a_n). \quad (7.6)$$

За теоремою Байєса:

$$\begin{aligned} v_0 &= \arg \max_{v \in V} \frac{P(a_1, a_2, \dots, a_n | x = v)P(x = v)}{P(a_1, a_2, \dots, a_n)} = \\ &= \arg \max_{v \in V} P(a_1, a_2, \dots, a_n | x = v)P(x = v) \end{aligned} \quad (7.7)$$

Оцінити  $P(x = v)$  можна легко; навіть за наявності невеликої кількості тестових прикладів можна оцінити частоту появи кожного із значень  $x$ . Але оцінити різні значення  $P(a_1, a_2, \dots, a_n | x = v)$  неможливо, оскільки їх кількість є занадто великою. Для того, щоб отримати кожну із цих ймовірностей, потрібно кожну комбінацію атрибутів спостерігати декілька раз, але це неможливо на практиці.

Тому наївний байєсівський класифікатор передбачає умовну незалежність атрибутів за умови заданого значення цільової функції:

$$P(a_1, a_2, \dots, a_n | x = v) = P(a_1 | x = v) \cdot P(a_2 | x = v) \cdot \dots \cdot P(a_n | x = v) \quad (7.8)$$



Всі параметри моделі можуть бути апроксимовані відносними частотами з множини навчальних даних. Недискретні атрибути повинні бути спочатку дискретизовані. Якщо даний клас і значення атрибута ніколи не зустрічалися разом у вибірці навчальних даних, то оцінка, яка ґрунтується на вірогідностях, буде дорівнювати нулю. Це проблема, оскільки при множенні нульова оцінка призведе до втрати інформації. Тому краще вводити невеликі поправки в усі імовірності таким чином, щоб жодна імовірність не була строго рівною нулю.

### **Алгоритм розпізнавання символів за допомогою найвішого байєсівського класифікатора**

Нехай  $\Pi_i$  – змінні ознак,  $i=1, \dots, n^2$ ,  $A$  – змінна “Значення символу”.  
Змінна  $A$  залежить від змінних  $\Pi_i$ , змінні  $\Pi_i$  – незалежні.

**Крок 1.** Вхідні зображення перетворюються у вектори розмірності  $n^2$ .

- а) Зображення розбивається на  $n$  клітинок у довжину та  $n$  клітинок у ширину.
- б) Приведення зображення до чисельного вигляду. Для кожної клітинки розраховується середнє значення яскравості.
- в) Дискретизація. Кожній клітинці ставиться у відповідність змінна ознаки  $\Pi_i$ ,  $i=1 \dots n^2$ .  $\Pi_i$  приймає значення 0, якщо середня яскравість клітинки перевищує деяке порогове значення (що відповідає білому кольору), або 1 у протилежному випадку (що відповідає чорному кольору).

**Крок 2.** Умовна ймовірність кожної ознаки знаходиться за формулою:

$$P(\Pi_i = p_i | A = a_j) = (n \lfloor A = a_j, \Pi_i = p_i \rfloor) / (n \lfloor A = a_j \rfloor), \quad (7.9)$$

де  $(n \lfloor A = a_j, \Pi_i = p_i \rfloor)$  – кількість появ конкретного значення  $p_i$  ознаки  $\Pi_i$  для букви  $a_j$ ,  
 $n \lfloor A = a_j \rfloor$  – кількість разів появи літери  $a_j$ .

**Крок 3.** Обчислення умовних ймовірностей усіх можливих значень змінної  $A$ . Згідно з теоремою Байєса, враховуючи незалежність ознак, умовна ймовірність дорівнює:

$$P(A = a_j | \Pi_1 = p_1, \Pi_2 = p_2, \dots, \Pi_{n^2} = p_{n^2}) = \frac{P(a_j) \prod_{i=1}^{n^2} P(\Pi_i = p_i | A = a_j)}{\sum_k P(a_k) \prod_{i=1}^{n^2} P(\Pi_i = p_i | A = a_k)} \quad (7.10)$$

Апріорні ймовірності  $P(a_j)$  припускаються рівними.

### **Методика розпізнавання символів за допомогою найвішої мережі Байєса із зворотним формуванням імовірнісного висновку**

Побудуємо байєсівську мережу такого вигляду: мережа складається з вершини “Значення символу”, яка впливає на вершини ознак (рис. 7.4).

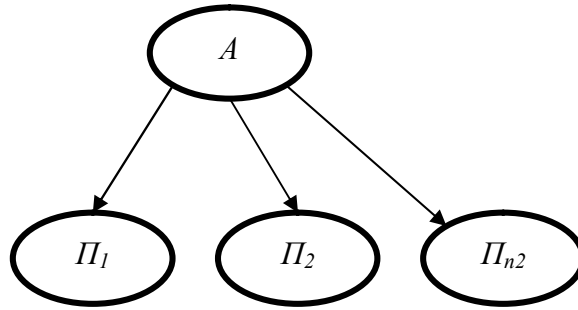


Рис. 7.4. Топологія наївної байєсівської мережі

На рис. 7.4 коренева вершина  $A$  відповідає значенню символу, а вершини ознак позначені як  $\Pi_i$ . Вершина  $A$  може набувати значення з множини  $\{a_j\} j=1\dots m$ .  $\Pi_i$  можуть набувати значення 0 або 1.

Алгоритм навчання та розпізнавання складається з таких кроків:

**Крок 1.** Вхідні зображення з множини навчальної вибірки та зображень для розпізнавання перетворюються у вектори розмірності  $n^2$ .

- a) Зображення розбивається на  $n$  клітинок у довжину та  $n$  клітинок у ширину.
- b) Приведення зображення до чисельного вигляду. Для кожної клітинки розраховується середнє значення яскравості.
- c) Дискретизація. Кожній клітинці ставиться у відповідність вершина ознаки  $\Pi_i, i=1\dots n^2$ .  $\Pi_i$  приймає значення 0, якщо середня яскравість клітинки перевищує деяке порогове значення (що відповідає білому кольору), або 1 в протилежному випадку (що відповідає чорному кольору).

**Крок 2.** Побудова мережі Байєса. Кожна з вершин ознак є залежною від кореневої вершини ‘Значення символу’. Коренева вершина може набувати значень з множини алфавіту.

**Крок 3.** Побудова таблиці спільного розподілу імовірностей для всієї мережі Байєса. Будується таблиця частот станів всієї мережі Байєса. За цією таблицею будується таблиця спільного розподілу імовірностей для всієї мережі.

**Крок 4.** На основі значень спільного розподілу імовірностей всієї мережі Байєса, за теоремою Байєса відбувається побудова таблиці умовних ймовірностей для всіх вершин мережі Байєса:

$$P(\Pi_i | A) = \frac{P(A | \Pi_i)}{P(A)} \tag{7.11}$$

**Крок 5.** Інстанціювання залежних вершин мережі Байєса відповідними значеннями для зображення, яке розпізнається.

$$P(\Pi_i = 0) = \begin{cases} 0, & 0, \text{ якщо } \Pi_i=1 \text{ для зображення, що розпізнається} \\ 1, & 1, \text{ якщо } \Pi_i=0 \text{ для зображення, що розпізнається} \end{cases} \tag{7.12}$$

$$P(\Pi_i = 1) = \begin{cases} 0, & 0, \text{ якщо } \Pi_i=0 \text{ для зображення, що розпізнається} \\ 1, & 1, \text{ якщо } \Pi_i=1 \text{ для зображення, що розпізнається} \end{cases} \tag{7.13}$$

**Крок 6.** Формування зворотного імовірнісного висновку. Будується модель множинної регресії. В якості регресорів беруть значення з таблиці умовних ймовірностей. Кількість регресорів співпадає з кількістю символів алфавіту. Залежною змінною є значення інстанційованих вершин. Невідомі параметри – це вірогідності кореневої вершини.

Модель множинної регресії у векторному вигляді має вигляд:

$$Y = X\theta, \tag{7.14}$$

$$\text{де } \theta = \begin{pmatrix} P(A = a_1) \\ \vdots \\ P(A = a_m) \end{pmatrix},$$

$$X = \begin{pmatrix} P(\Pi_i = 0 | A = a_1) & P(\Pi_i = 0 | A = a_2) & \dots & P(\Pi_i = 0 | A = a_m) \\ P(\Pi_1 = 0 | A = a_1) & P(\Pi_1 = 0 | A = a_2) & \dots & P(\Pi_1 = 0 | A = a_m) \\ P(\Pi_2 = 0 | A = a_1) & P(\Pi_2 = 0 | A = a_2) & \dots & P(\Pi_2 = 0 | A = a_m) \\ \vdots & \vdots & \ddots & \vdots \\ P(\Pi_{n^2} = 0 | A = a_1) & P(\Pi_{n^2} = 0 | A = a_2) & \dots & P(\Pi_{n^2} = 0 | A = a_m) \end{pmatrix},$$

$$Y = \begin{pmatrix} P(\Pi_1 = 0) \\ P(\Pi_1 = 0) \\ \vdots \\ P(\Pi_{n^2} = 0) \end{pmatrix}.$$

Для пошуку значень невідомих параметрів використовується рекурентний метод найменших квадратів. Отримані значення нормуються.

**Крок 7.** Серед можливих значень кореневої вершини обирається значення, яке має найбільшу імовірність.

### 7.6. Динамічна мережа Байєса

Динамічна мережа Байєса (ДМБ) – це мережа, у якій значення вузлів змінюються із часом. Їх перевага полягає в тому, що вони використовують табличне представлення умовних імовірностей, а це полегшує представлення різних нелінійних явищ [156].

Прикладом динамічної мережі Байєса є прихована модель Маркова, у якій на кожному шарі є один прихований вузол та один спостережуваний вузол. Ілюстрація моделі представлена на рис 7.5.

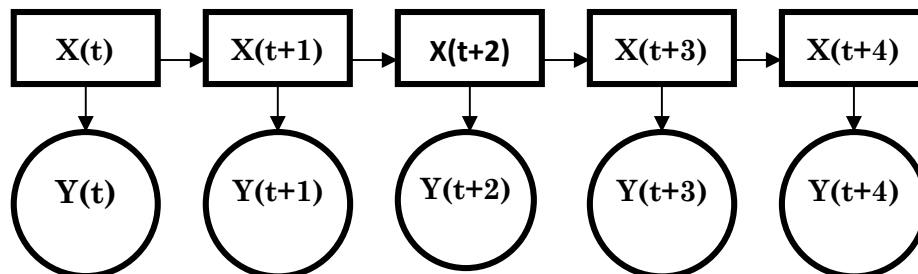


Рис. 7.5. Прихована двошарова модель Маркова, в якій X – приховані дискретні, а Y – дискретні або неперервні спостережувані вузли

На рис. 7.5 круглі вершини позначають безперервні вузли, квадратні позначають дискретні.  $X$  – приховані вузли, а  $Y$  – спостережувані. Для задавання динамічної мережі Байєса потрібно визначити початковий розподіл, топологію усередині шару і міжшарову топологію (між двома шарами). Мережі такого виду використовують при розпізнаванні мовних сигналів. У цьому випадку вузли  $Y(t), Y(t+1), Y(t+2), \dots$  являють собою фонему при вимові слів, а вузли – це букви, з яких складається вимовлене слово. Така модель буде динамічною у тому розумінні, що дана мережа буде представляти собою безліч повторюваних блоків у послідовні моменти часу.

Марковський ланцюг задається початковим розподілом ймовірностей  $p^o(x)$  і ймовірностями переходу  $T(x'; x)$ .  $T(x'; x)$  – це розподіл наступного елемента ланцюга в залежності від попереднього. Розподіл на  $(t+1)$ -у кроці дорівнює:

$$p_{t+1}(x') = \int T(x'; x) p^o(x) dx. \quad (7.15)$$

У дискретному випадку  $T(x'; x)$  – це матриця ймовірностей переходів між станами  $p(x' = i | x = j)$ .

Будемо розглядати тільки дискретні задачі. За допомогою марковської моделі можна моделювати відповідний процес, спостерігаючи при цьому деякі функції від цього процесу.

Приховані марковські моделі припускають, що зазвичай не можна отримати самі стани, оскільки не відомо, скільки є цих станів і які між ними зв'язки: це невідомі параметри моделі.

Відомі лише спостережувані величини  $y(t)$ , які залежать від прихованих змінних  $x(t)$ . Завдання полягає в тому, щоб визначити приховані параметри процесу. Іншими словами, за наявними даними  $y(t)$  необхідно визначити, які найбільш імовірні  $x(t)$  і найбільш імовірна модель цього марковського процесу. Головна властивість цієї моделі така: наступний стан залежить тільки від попереднього і не залежить від історії. Формально це означає, що імовірність  $x(t)$  за умови всіх інших установлених значень дорівнює імовірності  $x(t)$  за умови тільки попереднього значення:

$$p(x(t) = x_j | x(t-1) = x_{j_{t-1}}, \dots, x(1) = x_{j_1}) = p(x(t) = x_j | x(t-1) = x_{j_{t-1}}). \quad (7.16)$$

Більше того, імовірності  $a_{ij} = p(x(t) = x_j | x(t-1) = x_{j_{t-1}})$  від часу  $t$  не залежать. Ці ймовірності складають матрицю переходу  $A = (a_{ij})$ .

Для ймовірностей переходу  $(a_{ij})$  виконуються такі властивості:

1.  $a_{ij} \geq 0$ ;
2.  $\sum_j a_{ij} = 1$  (якщо почнемо рух із стану, і куди-небудь точно потрапимо).

### **Основні задачі при моделюванні моделей Маркова**

При моделюванні моделей Маркова необхідно вирішити три основні задачі моделювання:

1. Знайти ймовірність послідовності спостережень у даній моделі.

2. Знайти “оптимальну” послідовність станів за умови даної моделі і даної послідовності спостережень.

3. Знайти найбільш правдоподібну модель (параметри моделі).

Нехай, маємо:

$X = \{x_1 \dots x_n\}$  – множина станів;

$V = \{v_1 \dots v_m\}$  – алфавіт, з якого обирають спостережувані  $y$  (множина значень  $y$ );

$q_t$  – стан у момент часу  $t$ ;

$y_t$  – спостереження в момент часу  $t$ ;

Дані позначимо через  $D = d_1 \dots d_T$

Для ймовірностей і розподілів будемо використовувати такі позначення:

$a_{ij} = p(q_{i+1} = x_j | q_i = x_i)$  – імовірність переходу із стану  $i$  в стан  $j$ ;

$b_j(k) = p(v_k | x_j)$  – імовірність отримати дані  $v_k$  в стані  $j$ ;

Початковий розподіл:  $\pi = \{\pi_j\}, \pi_j = p(q_1 = x_j)$ .

Модель  $\lambda = (A, B, \pi)$  складається з матриці переходу  $A$ , матриці спостережень  $B$  та початкового розподілу  $\pi$ .

### **Пошук ймовірності послідовності спостережень в заданій моделі**

За заданою моделлю  $\lambda = (A, B, \pi)$  та послідовністю спостережень  $D$  необхідно знайти  $p(D | \lambda)$ . Фактично, для цього потрібно оцінити, наскільки добре модель підходить для даних.

Крім явних параметрів, є неявні приховані параметри  $Q$ , знайдемо суму:

$$p(D | \lambda) = \sum_Q p(D | Q, \lambda) p(Q | \lambda). \quad (7.17)$$

Імовірність  $p(D | \lambda)$  послідовності  $Q = q_1 \dots q_T$  за умови  $\lambda$  – це імовірність того, що оберемо перший елемент, а потім будемо переходити від першого до другого, від другого до третього і так далі. Для того, щоб знайти імовірність  $D$  за умови послідовності  $Q$ , на кожному кроці знаходимо імовірності отримання відповідних даних. Кожна послідовність – це окрема подія. Вони не перетинаються, тому можна розрахувати суму по всіх подіях:

$$p(D | \lambda) = \sum_Q p(D | Q, \lambda) p(Q | \lambda) = \sum b_{q_1}(d_1) \cdot b_{q_2}(d_2) \pi_{q_1} a_{q_1 q_2} \dots a_{q_{T-1} q_T} \quad (7.18)$$

Скористаємося так званою forward-backward процедурою, по суті – обчисленням на решітці, як у декодуванні. Будемо послідовно обчислювати проміжні величини виду:

$$\alpha_t(i) = p(d_1 \dots d_t, q_t = x_i | \lambda), \quad (7.19)$$

тобто шукані ймовірності, але ще із урахуванням поточного стану. Величина  $\alpha_t(i)$  показує імовірність усіх наявних даних аж до моменту  $t$  і того, у який прийшли в стан  $x_i$  на кроці  $t$  в даній моделі.

Тоді шукана ймовірність  $p(D | \lambda)$  представляється у вигляді:

$$p(D | \lambda) = \sum_{i=1}^n \alpha_T(i) \quad (7.20)$$

Допоміжні величини  $\alpha_t(i)$  можна обчислити за методом динамічного програмування. Спочатку ініціалізуємо  $\alpha_1(i)$  добутком початкового розподілу  $\pi_i$  та ймовірності отримати  $d_1$ . Потім виконуємо крок індукції. Для обчислення  $\alpha_{t+1}(i)$  необхідно підрахувати суму по всіх попередніх станах ймовірностей, досягти цього стану й перейти в стан  $j$ ; після цього потрібно помножити цю суму на ймовірність одержати в  $j$ -му стані нові дані  $d_{t+1}$ . Після того, як дійдемо до кроку  $T$ , просумуємо і знайдемо величину  $p(D | \lambda)$ .

Фактично, це тільки прямий прохід, зворотний нам тут не потрібен. Зворотний прохід обчислював би умовні ймовірності  $\beta_t(i) = p(d_{t+1} \dots d_T | q_t = x_i, \lambda)$  – ймовірності одержання ланцюжка  $d_{t+1} \dots d_T$ , починаючи зі стану  $i$  на кроці  $t$ . Їх можна обчислити шляхом ініціалізації  $\beta_T(i) = 1$ , а потім за індукцією отримуємо:

$$\beta_t(i) = \sum_{j=1}^n a_{ij} b_j(d_{t+1}) \beta_{t+1}(j). \quad (7.21)$$

Ці величини знадобляться при розв'язанні другої і третьої задач.

### Пошук “оптимальної” послідовності станів моделі

При розв'язку задачі можливі два варіанти:

- розв'язувати “побітово”, відповідаючи на запитання, який найбільш ймовірний стан у момент часу  $j$ .
- розв'язувати задачу знаходження найбільш ймовірної послідовності станів.

Ці варіанти розв'язання задачі принципово відрізняються один від одного. Якщо взяти найбільш ймовірні стани в момент часу  $j$  і скласти з них послідовність, то можемо отримати некоректну послідовність (містить перехід, ймовірність якого рівна 0).

Наведемо побітовий розв'язок задачі. Для цього слід ввести допоміжні змінні:

$$\gamma_t(i) = p(q_t = x_i | D, \lambda). \quad (7.22)$$

$\gamma_t(i)$  – це ймовірність того, що прийдемо на  $t$ -му кроці в стан  $i$  за умови заданих даних та заданої моделі. Задача полягає в тому щоб знайти

$$q_t = \arg \max_{1 \leq i \leq n} \gamma_t(i), \quad 1 \leq t \leq T. \quad (7.23)$$

Для знаходження значення  $q_t$ , яке максимізує  $\gamma_t(i)$ , застосуємо варіант *min-max* алгоритму.

Виражаємо  $\gamma_t(i)$  через  $\alpha$  і  $\beta$ :

$$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{p(D | \lambda)} = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^n \alpha_t(i) \beta_t(i)} \quad (7.24)$$

Знаменник можна не враховувати, тому що потрібен тільки  $\arg \max$ .

Для того щоб дати відповідь на запитання про найбільш імовірну послідовність, будемо використовувати алгоритм динамічного програмування Вітербі.

Введемо допоміжні змінні:

$$\delta_i(i) = \max_{q_1, \dots, q_{t-1}} p(q_1 q_2 \dots q_t = x_i, d_1 d_2 \dots d_t | \lambda) \quad (7.25)$$

$\delta_i(i)$  – максимальна імовірність досягти стан  $x_i$  на кроці  $t$  серед всіх шляхів із заданими спостереженнями.

За індукцією отримаємо:

$$\delta_{t+1}(j) = \left[ \max_i \delta_i(i) a_{ij} \right] b_j(d_{t+1}). \quad (7.26)$$

Крім того, знадобиться запам'ятовувати аргументи, а не лише значення; для цього використовується масив  $\psi_i(i)$ .

Спочатку ініціалізуємо  $\delta_i(i)$  початковими ймовірностями, а  $\psi_i(i)$  – порожнім масивом. Потім за індукцією обчислюємо значення  $\delta_i(i)$  і аргументи  $\psi_i(i)$ , на яких досягаються максимуми. Коли дійдемо до кроку  $T$ , підрахуємо  $p^*$  і  $q_T^*$ . Після цього, повертаючись назад по масиву  $\psi$ , будемо обчислювати аргументи.

### ***Пошук найбільш правдоподібної моделі***

Задача полягає в знаходженні параметрів моделі  $a_{ij}$ ,  $b_j(k)$ ,  $\pi_i$  за умови наявних даних. Необхідно відзначити, що знайти максимум аналітично  $P(D|\lambda)$  неможливо. Розглянемо ітераційну процедуру (по суті – градієнтний підйом), яка приведе до локального максимуму цієї функції. У функції може бути декілька локальних максимумів, тому на практиці цей алгоритм слід запускати декілька разів, щоб він попадав у різні елементи простору і міг би привести до різних максимумів. Даний алгоритм називають алгоритмом Баума-Велча (Baum-Welch). Насправді – це окремий випадкового алгоритму максимізації математичного сподівання. За наявності деякої моделі з деякими параметрами, згідно з ідеєю алгоритму ЕМ, спочатку знаходимо математичне очікування даних за умови цієї моделі, потім корегуємо модель за умови наявних даних. У ході такого ітеративного процесу ми приходимо до найкращої моделі.

Вводимо допоміжні змінні – імовірності того, що в момент часу  $t$  знаходимось у стані  $x_i$ , а в момент часу  $t+1$  – в стані  $x_j$ :

$$\xi_1(i, j) = p(q_t = x_i, q_{t+1} = x_j | D, \lambda) \quad (7.27)$$

Вираз для  $\xi_1(i, j)$  можна переписати таким чином:

$$\xi_1(i, j) = \frac{\alpha_t(i) a_{ij} b(d_{t+1}) \beta_{t+1}(j)}{p(D|\lambda)} = \frac{\alpha_t(i) a_{ij} b(d_{t+1}) \beta_{t+1}(j)}{\sum_i \sum_j \alpha_t(i) a_{ij} b(d_{t+1}) \beta_{t+1}(j)}. \quad (7.28)$$

Відзначимо також, що  $\gamma_t(i) = \sum_j \xi_t(i, j)$ .

$\sum_i \gamma_i(i)$  – це кількість переходів, що очікуються із стану  $x_i$ , а  $\sum_j \xi_i(i, j)$  – із  $x_i$  в стан  $x_j$ .

На кроці  $M$  будемо переоцінювати такі імовірності:

$$\bar{\pi}_i = \gamma_i(i) \quad (7.29)$$

$$\bar{a}_{ij} = \frac{\sum_j \xi_i(i, j)}{\sum_i \gamma_i(i)}, \quad (7.30)$$

$$\bar{b}_i(k) = \frac{\sum_j \Delta(d_i, v_k) \gamma_i(i)}{\sum_i \gamma_i(i)} \quad (7.31)$$

Алгоритм розпочинають з деякої випадкової моделі  $\lambda = (A, B, \pi)$ , потім обчислюють допоміжні змінні, підраховують  $\lambda = (A, B, \pi)$  і знову перераховують параметри і т. д.

### Обґрунтування алгоритму Баума-Велча

Відстань Кульбака-Лейблера (Kullback-Leibler distance, divergence) – це інформаційно-теоретична міра того, наскільки далекі розподіли один від одного:

$$DKL(p_1, p_2) = \sum_x p_1(x) \log \frac{p_1(x)}{p_2(x)} \quad (7.32)$$

Відомо, що ця відстань завжди невід’ємна і дорівнює нулю тоді і тільки тоді,  $p_1$  і  $p_2$  збігаються в кожній точці:  $p_1 \equiv p_2$ .

Знайдемо розподіли  $p_1$  та  $p_2$  на множині послідовних станів для моделей  $\lambda$  і  $\lambda'$  таким чином:

$$p_1(Q) = \frac{p(Q, D | \lambda)}{p(D | \lambda)}, \quad (7.33)$$

$$p_1(Q) = \frac{p(Q, D | \lambda')}{p(D | \lambda')}. \quad (7.34)$$

Тепер відстань Кульбака-Лейблера для розподілу  $p_1$  і  $p_2$ :

$$\begin{aligned} 0 \leq D_{LK}(\lambda, \lambda') &= \sum_Q p_1(Q) \log \frac{p_1(Q)}{p_2(Q)} = \sum_Q \frac{p_1(Q, D | \lambda)}{p_2(D | \lambda)} \log \frac{p(Q, D | \lambda) p(D | \lambda')}{p(Q, D | \lambda') p(D | \lambda)} = \\ &= \log \frac{p(D | \lambda')}{p(D | \lambda)} + \sum_Q \frac{p(Q, D | \lambda)}{p(D | \lambda)} \log \frac{p(Q, D | \lambda)}{p(D | \lambda')} \end{aligned} \quad (7.35)$$

Введемо допоміжну функцію:

$$Q(\lambda, \lambda') = \sum_Q p(Q | D, \lambda) \log p(Q | d, \lambda').$$



З нерівності випливає, що

$$\frac{Q(\lambda, \lambda') - Q(\lambda, \lambda)}{p(D | \lambda)} \leq \log \frac{p(D | \lambda')}{p(D | \lambda)}. \quad (7.36)$$

Якщо  $Q(\lambda, \lambda') > Q(\lambda, \lambda)$ , то  $0 \leq \log \frac{p(D | \lambda')}{p(D | \lambda)}$ , відповідно  $p(D | \lambda') > p(D | \lambda)$ .

Отже, якщо максимізуємо  $Q(\lambda, \lambda')$  по  $\lambda'$ , то тим самим будемо рухатися у потрібний бік.

Таким чином, потрібно максимізувати  $Q(\lambda, \lambda')$  по  $\lambda'$ . Перепишемо:

$$\begin{aligned} Q(\lambda, \lambda') &= \sum_Q p(Q | D, \lambda) \log p(Q | D, \lambda') = \sum_Q p(Q | D, \lambda) \log \pi_{q_1} \prod_t a_{q_{t-1} q_t} b_{q_t}(d_t) = \\ &= \sum_Q p(Q | D, \lambda) \log \pi_{q_1} + \sum_Q p(Q | D, \lambda) \sum_Q \log a_{q_{t-1} q_t} + \sum_Q p(Q | D, \lambda) \sum_Q \log b_{q_t}(d_t). \end{aligned} \quad (7.37)$$

Останній вираз можна легко диференціювати по  $a_{ij}, b_j(k)$  і  $\pi_i$ , додати відповідні множники Лагранжа і розв'язати. Оскільки параметри  $a_{ij}, b_j(k)$  і  $\pi_i$ , по яких виконується оптимізація, входять у кожен із трьох доданків незалежно, то для  $Q(\lambda, \lambda')$  можна оптимізувати кожен із цих доданків окремо.

Розглянемо перший доданок:

$$\sum_Q p(Q | D, \lambda) \log \pi_{q_1} = \sum_{i=1}^n p(q_1 = x_i | D, \lambda) \log \pi_i \quad (7.38)$$

Додаючи множник Лагранжа  $\mu$  і використовуючи обмеження  $\sum_i \pi_i = 1$ , запишемо вираз для похідної та прирівняємо його до нуля:

$$\frac{\partial}{\partial \pi_i} \left( \sum_{i=1}^n p(q_1 = x_i | D, \lambda) \log \pi_i + \mu \left( \sum_{i=1}^n \pi_i - 1 \right) \right) = 0 \quad (7.39)$$

Тепер диференціюємо цей вираз по  $\pi_i$ , знаходимо суму по  $i$  та скористаємось рівністю  $\sum_i \pi_i = 1$  для того, щоб знайти  $\mu$ ; і отримуємо розв'язок відносно  $\pi_i$ :

$$\pi_i = \frac{p(q_1 = x_i | D, \lambda)}{p(D | \lambda)} \quad (7.40)$$

Перепишемо другий доданок суми  $Q(\lambda, \lambda')$ :

$$\sum_Q p(Q | d, \lambda) \sum_t \log a_{q_{t-1} q_t} = \sum_{i=1}^n \sum_{j=1}^n \sum_t p(q_t = x_i, q_{t+1} = x_j | D, \lambda) \log a_{ij}. \quad (7.41)$$

Вводячи множник Лагранжа та застосовуючи обмеження  $\sum_j a_{ij} = 1$ , отримуємо

$$\bar{a}_{ij} = \frac{\sum_t p(q_t = x_i, q_{t+1} = x_j | D, \lambda)}{\sum_t p(q_t = x_i | D, \lambda)} = \frac{\sum_j \xi_t(i, j)}{\sum_t \gamma_t(i)}. \quad (7.42)$$

Третій доданок суми  $Q(\lambda, \lambda')$  запишемо в вигляді

$$\sum_Q p(Q | d, \lambda) \sum_t \log b_{q_t}(d_t) = \sum_{i=1}^n \sum_t p(q_t = x_i | D, \lambda) \log b_i(d_t). \quad (7.43)$$

Знову застосуємо метод множників Лагранжа з обмеженням  $\sum_j b_j(j) = 1$ .

Відмітимо, що лише спостереження  $d_t$ , яке дорівнює  $v_k$ , вносить вклад в остаточне значення імовірності:

$$\bar{b}_i(k) = \frac{\sum_t \Delta(d_t, v_k) p(q_t = x_i | D, \lambda)}{\sum_t p(q_t = x_i | D, \lambda)} = \frac{\sum_j \Delta(d_t, v_k) \gamma_t(i, j)}{\sum_t \gamma_t(i)}. \quad (7.44)$$

### Нормально розподілені випадкові величини

При аналізі та моделюванні випадкових величин найчастіше використовують нормальний або Гаусів розподіл, який має декілька властивостей, що сприяють прискоренню та спрощенню аналізу процесів, він описується виразом:

$$N(x; \mu_x, \sigma_x) = \frac{1}{\sqrt{(2\pi\sigma_x)}} \cdot \exp\left(-\frac{1}{2} \frac{(x - \mu_x)^2}{\sigma_x}\right), \quad (7.45)$$

де  $\mu_x$  – середнє значення процесу;  $\sigma_x$  – стандартне відхилення.

Багатовимірний нормальний розподіл у теорії ймовірностей – це узагальнення одномірного нормального розподілу.

Випадковий вектор  $X = (x_1, x_2, \dots, x_n)^T : \Omega \rightarrow R^n$  має багатовимірний нормальний розподіл, якщо виконуються одна з таких еквівалентних умов:

1. Довільна лінійна комбінація компонентів вектора  $\sum_{i=1}^n a_i x_i$  має нормальний розподіл або є константою.

2. Існує вектор незалежних стандартних нормальних випадкових величин  $Z = (z_1, \dots, z_m)^T$ , дійсний вектор  $\mu = (\mu_1, \dots, \mu_m)^T$  і матриця  $A$  розмірності  $n \times m$ , такі що:

$$X = AX + \mu. \quad (7.46)$$

3. Існує вектор  $\mu \in R^n$  та невід’ємно визначена симетрична матриця  $\sum$  розмірності  $n \times n$  такі, що густина розподілу імовірності вектора  $X$  має вигляд:

$$f_X(x) = \frac{1}{(2\pi)^{\frac{n}{2}} |\sum|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T \sum^{-1}(x-\mu)} \quad x \in R^n \quad (7.47)$$

4. Існує вектор  $\mu \in R^n$  і невід’ємно визначена симетрична матриця  $\sum$  розмірності  $n \times n$ , такі що характеристична функція вектора  $X$  має вигляд:

$$\Phi_X(u) = e^{i\mu^T u - \frac{1}{2}u^T \sum u}, \quad u \in R^n. \quad (7.48)$$

Очевидно, що для опису випадкових процесів можна скористатись будь-яким іншим розподілом, який краще відповідає реальності. Іншими типами розподілів

можуть бути такі: бета-розподіл, біноміальний, розподіл  $\chi$  - квадрат, нецентральний розподіл  $\chi$  - квадрат, дискретний рівномірний розподіл, експоненціальний,  $F$  - розподіл, нецентральний  $F$ - розподіл, гамма-розподіл, геометричний, гіпергеометричний, логнормальний, негативний біноміальний та розподіли Пуассона, Рейлі,  $t$  - розподіл Стьюдента, нецентральний  $t$  - розподіл, рівномірний неперервний розподіл та розподіл Вейбулла.

**Алгоритм Баума-Велча для випадку неперервних спостережуваних величин**

Введемо наступні позначення:  $N$  – загальна кількість станів у моделі;  $S=\{S_1, \dots, S_N\}$  – сукупність можливих станів моделі; поточний стан у момент часу  $t - q_t$ ;  $A=\{a_{ij}\}$  – матриця переходів, де  $a_{ij} = P[q_{t+1} = S_j | q_t = S_i]$ ;  $B = \{b_1(0), \dots, b_N(0)\}$  – множина функцій розподілу ймовірностей, які описують розподіл імовірності  $b_j(0)$ ,  $1 \leq j \leq N$ , для стану  $S_j$ , де  $o_t$  – спостереження в момент часу  $t$  з послідовності спостережень  $o = \{o_1, \dots, o_T\}$ . Розподіл описується нормальним законом розподілу з  $\pi = \{\pi_i\}$  – таблиця ймовірностей початкового стану, де  $\pi_i = P\{q_1 = S_i\}$ ,  $1 \leq i \leq N$ . Набір параметрів  $\lambda = (A, B, \pi)$  повністю задає мережу.

Пряма процедура алгоритму Баума-Велча:

$$\alpha_i(1) = \pi_i(1), \quad i = 1, \dots, N; \tag{7.49}$$

$$\alpha_j(t+1) = b_j(o_{t+1}) \sum_{i=1}^N \alpha_i(t) a_{ij}, \quad j = 1, \dots, N, \quad t = 1 \dots T. \tag{7.50}$$

Зворотна процедура:

$$\beta_j(T) = 1, \quad j = 1, \dots, N; \tag{7.51}$$

$$\beta_j(t) = \sum_{j=1}^N \beta_j(t+1) a_{ij} b_j(o_{t+1}), \quad j = 1, \dots, N, \quad t = 1 \dots T. \tag{7.52}$$

Допоміжні змінні:

$$\gamma_i(t) = \frac{\alpha_i(t) \beta_i(t)}{\sum_{j=1}^N \alpha_j(t) \beta_j(t)}, \quad i = 1, \dots, N, \quad t = 1 \dots T \tag{7.53}$$

$$\xi_{ij}(t) = \frac{\alpha_i(t) a_{ij} \beta_j(t+1) b_j(o_{t+1})}{\sum_{i=1}^N \sum_{j=1}^N \alpha_i(t) a_{ij} \beta_j(t+1) b_j(o_{t+1})}, \quad i = 1, \dots, N, \quad j = 1, \dots, N, \quad t = 1 \dots T \tag{7.54}$$

Нове значення початкового розподілу:

$$\bar{\pi}_i = \gamma_i(1), \quad i = 1 \dots N \tag{7.55}$$

Нове значення матриці переходів:

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}, \quad i, j = 1 \dots N \tag{7.56}$$

Нове значення вектору математичного сподівання функцій розподілу спостережень для кожного прихованого стану  $j$ :

$$\hat{b}_j(o) \rightarrow \sigma_j = \frac{\sum_{t=1}^T o_t \gamma_j(t)}{\sum_{t=1}^T \gamma_j(t)}, \quad j = 1 \dots N \quad (7.57)$$

Нове значення коваріаційної матриці:

$$\hat{b}_j(o) \rightarrow \sigma_j = \frac{\sum_{t=1}^T (o_t - \mu_j)^T \gamma_j(t) (o_t - \mu_j)}{\sum_{t=1}^T \gamma_j(t)}, \quad j = 1 \dots N \quad (7.58)$$

### Контрольні питання

1. У чому полягає задача розпізнавання образів? Дайте означення та наведіть приклади.
2. Які характерні риси притаманні задачам розпізнавання образів?
3. Які методи розпізнавання образів використовуються?
4. Надайте класифікацію методів розпізнавання образів.
5. Що таке інтенціональні та екстенціональні методи розпізнавання образів? В чому полягає між ними різниця?
6. Які евристичні методи розпізнавання образів використовуються? Наведіть приклад.
7. Які математичні методи розпізнавання образів використовуються? Наведіть приклад.
8. Які лінгвістичні методи розпізнавання образів використовуються? Наведіть приклад.
9. Надайте класифікацію задач розпізнавання.
10. Що таке наївний байєсівський класифікатор, його роль та місце серед методів розпізнавання? Приклади використання.
11. Дайте означення моделі наївного байєсівського класифікатора.
12. Поясніть по кроках алгоритм наївного байєсівського класифікатора.
13. Назвіть особливості динамічної мережі Байєса.
14. Назвіть послідовність дій при пошуку “оптимальної” послідовності станів моделі.
15. Що таке нормально розподілені випадкові величини?
16. Для чого призначений алгоритм Баума-Велча.

## РОЗДІЛ 8 СИСТЕМА РОЗПІЗНАВАННЯ СИМВОЛІВ НА ОСНОВІ ДИНАМІЧНОЇ МЕРЕЖІ БАЙЄСА

- Алгоритм розпізнавання текстових символів на основі мереж Байєса
- Побудова структури програмної реалізації системи розпізнавання символів
- Емпіричне оцінювання алгоритмів розпізнавання
- Огляд комп'ютерних систем для розпізнавання символів

### 8.1. Опис алгоритму розпізнавання текстових символів на основі мереж Байєса

Алгоритм розпізнавання символів на основі мереж Байєса складається з наступних кроків.

**Крок 1.** Представлення зображень з навчальної та тестової вибірки у вигляді множини спостережень.

- а) Зображення розбивається на  $d$  клітинок у довжину та  $d$  клітинок у ширину.
- б) Для кожної клітинки розраховується середнє значення яскравості.
- в) Нормалізація середнього значення яскравості.
- г) Спостереження представляють собою потік з  $T=d$  рядків нормалізованих середніх значень яскравості клітинок.

На рис 8.1. розглянуто приклад побудови множини спостережень.

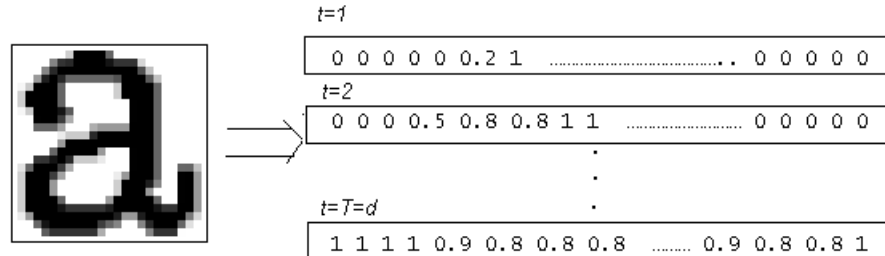


Рис 8.1. Приклад побудови множини спостережень для зображення літери “А”

Зазначимо, що кількість спостережень збігається з розмірністю спостережуваного вектора.

**Крок 2.** Для кожного символу будується окрема мережа Байєса.

- а) Вектор початкового розподілу  $\pi$ , матриця переходів  $A$ , параметри функцій розподілу для кожного прихованого стану  $\mu$  та  $\sigma$  ініціалізуються деякими випадковими значеннями.

б) Нормалізація параметрів.

**Крок 3.** Вибір навчальної послідовності спостережень для першого символу.

**Крок 4.** Ітерація алгоритму Баума-Велча:

- а) Обчислення значень  $\alpha_j(t)$ ,  $i, j=1..N$ ,  $t=1..T$  за допомогою прямої процедури за формулами (3.44, 3.45).
- б) Обчислення значень  $\beta_j(t)$ ,  $i, j=1..N$ ,  $t=1..T$ , за допомогою зворотної процедури за формулами (3.46, 3.47).

в) За формулами (3.48, 3.49) знаходять значення допоміжних змінних  $\gamma_i(t)$ ,  
 $i, j = 1 \dots N$ ,  $\xi_{ij}(t)$ ,  $j = 1 \dots N$ ,  $t = 1 \dots T$

г) Обчислення нових значень параметрів моделі  $\pi, A, \mu, \sigma$ .

**Крок 5.** Перевірка виконання навчання на всіх навчальних послідовностях спостережень даного символу. Якщо для даної мережі символу є послідовності, на яких не виконувалось навчання, то виконується вибір нової навчальної послідовності та перехід на крок 4. Інакше переходимо на крок 6.

**Крок 6.** Перевірка завершення алгоритму Баума-Велча. Розраховується значення  $\sum_{i=1}^N \alpha_T(i) = p$  та порівнюється з попереднім значенням  $p'$ . Якщо  $|p - p'| < eps$ , то переходимо на крок 7. Інакше переходимо на крок 4 та повторюємо навчання на всіх навчальних множинах спостережень для даного символу.

**Крок 7.** Якщо виконано навчання мережі Байєса для кожного символу, то навчання завершено. Інакше обираємо мережу Байєса для наступного символу та переходимо на крок 4.

**Крок 8.** Розпізнавання. Множину спостережень з тестової вибірки пред'являють мережі Байєса кожного символу. Знаходяться величини  $\sum_{i=1}^N \alpha_T^k(i) = p^k$ , де верхній індекс  $k$  відповідає номеру мережі Байєса. Вважаємо, що дану послідовність згенерувала мережа Байєса, яка дала найбільше значення  $p^k$ , тобто

$$k^* = \arg \max_k \sum_{i=1}^N \alpha_T^k(i) = \arg \max_k p^k .$$

Отже, розглянувши методику розпізнавання символів на основі мереж Байєса, можна зробити такі висновки. Наївний Байєсівський класифікатор є простим імовірнісним класифікатором, що ґрунтується на застосуванні теореми Байєса зі строгим припущенням про незалежність. Незважаючи на дуже грубе припущення, цей класифікатор може працювати досить ефективно. У випадку наївної мережі Байєса зі зворотним формуванням ймовірнісного висновку також було використано припущення про незалежність ознак. Але ймовірнісний висновок тут будується на основі навчальних даних із застосуванням моделі множинної регресії та рекурентного методу найменших квадратів. Динамічна мережа Байєса, що використовується в даному випадку для розпізнавання символів, представляє зображення як потік даних, а саме вертикальний потік рядків. Мережа складається з двох вузлів, значення яких може змінюватись із часом.

Дискретний прихований вузол (прихований стан мережі) впливає на неперервний вузол (спостереження). Попереднє значення прихованого стану впливає на його наступне значення. Параметри мережі є невідомими та знаходяться за допомогою методу максимізації математичного сподівання. Спостереженнями в даній моделі є рядки зображення літери. Крім розпізнавання літер, приховані моделі Маркова можуть застосовуватися для розв'язання інших задач розпізнавання образів, наприклад, для розпізнавання мови.

## 8.2. Приклад побудови структури програмної реалізації системи розпізнавання символів на основі мереж Байєса

Структура програмної реалізації системи розпізнавання символів на основі мереж Байєса має свої особливості. Головними блоками програми є такі (рис. 8.2):

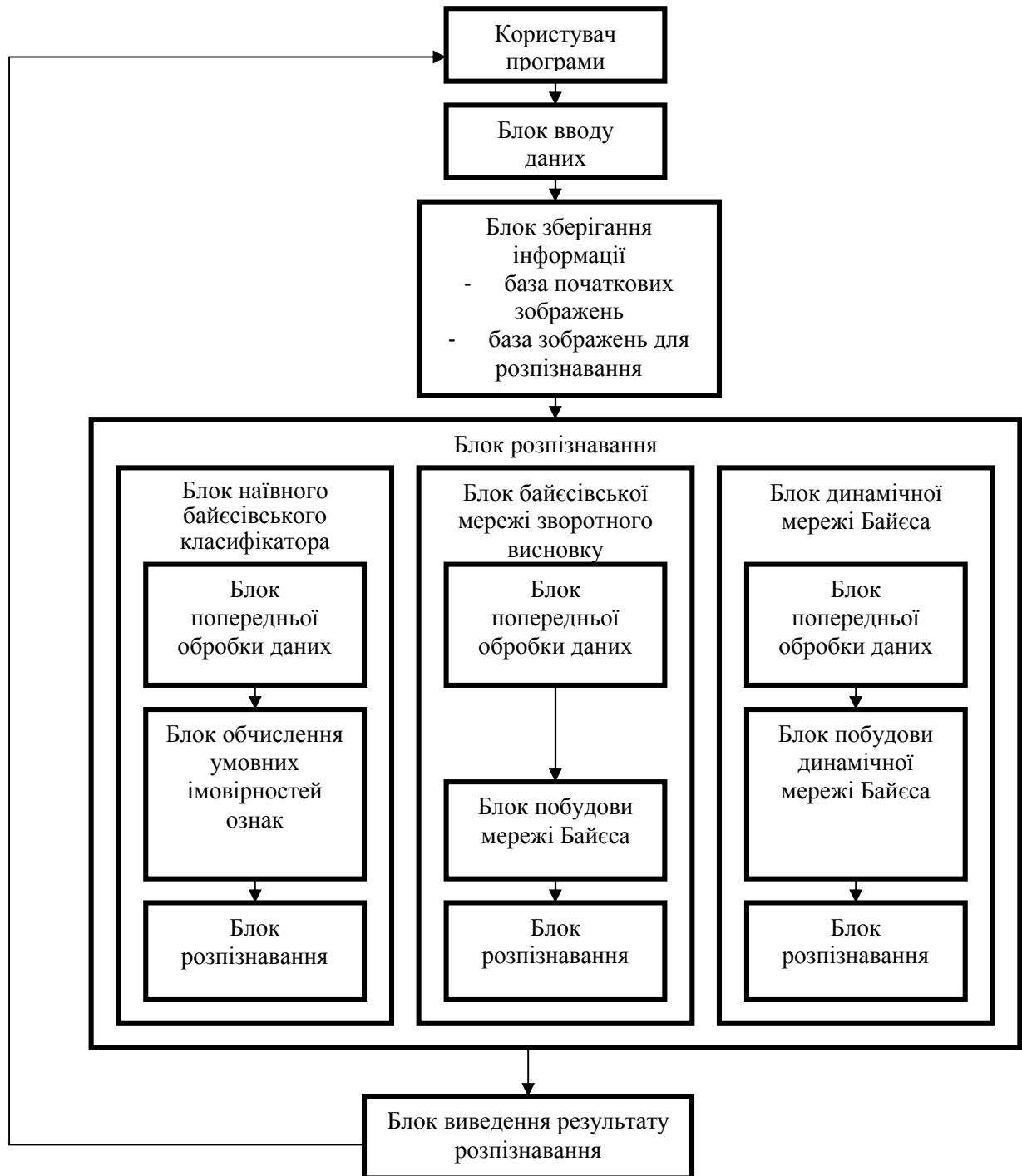


Рис 8.2. Спрощена архітектура програмної реалізації системи розпізнавання символів на основі мереж Байєса

- блок вводу даних;
- блок зберігання інформації;
- блок розпізнавання;
- блок виводу результату розпізнавання.

У свою чергу, блок розпізнавання складається з блоку наївного байєсівського класифікатора, блоку наївної мережі Байєса із зворотним формуванням імовірнісного висновку та блоку динамічної байєсівської мережі. Представлена на рис. 8.3 структура описує блок мережі Байєса зворотного імовірнісного висновку.

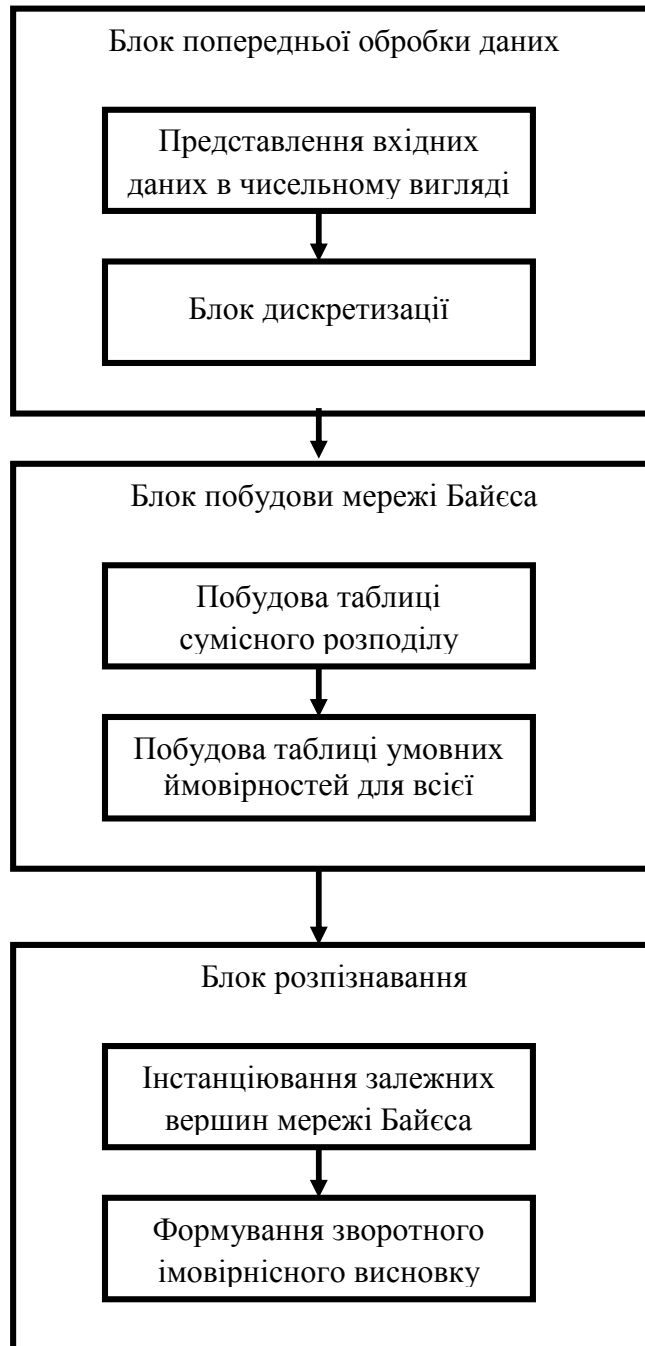


Рис. 8.3. Структура блоку мережі Байєса зворотного висновку



На основі даної структури авторами було запропоновано архітектуру пристрою для розпізнавання символів.

Блок наївної байєсівської мережі із зворотним формуванням імовірнісного висновку складається з блоку попередньої обробки даних, блоку побудови мережі Байєса та блоку розпізнавання.

На рис. 8.4 представлена структура блоку динамічної байєсівської мережі.

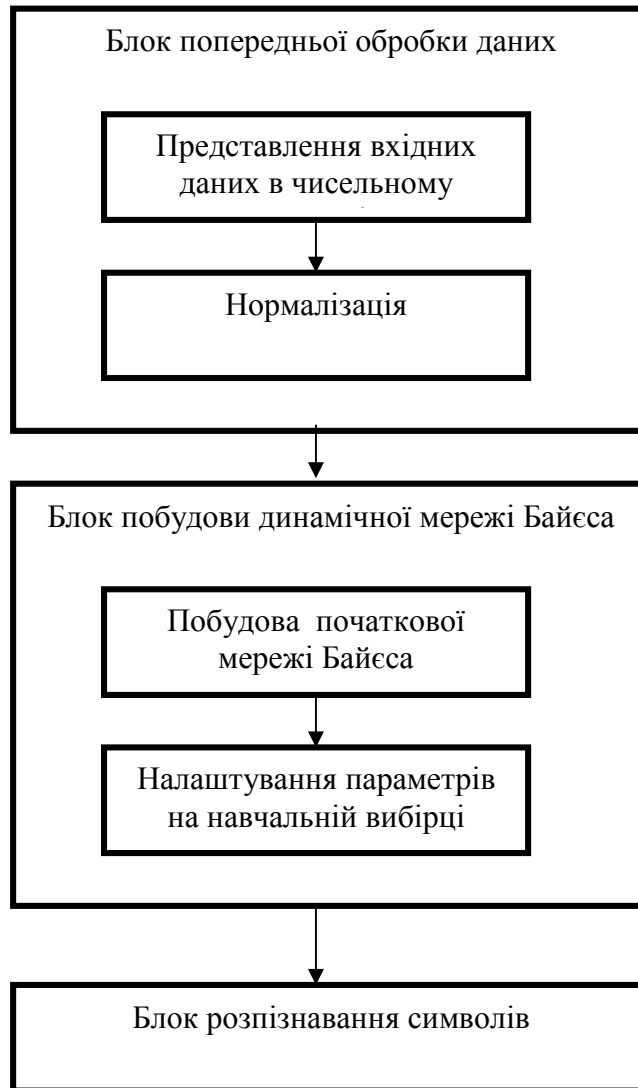


Рис. 8.4. Структура блоку динамічної мережі Байєса

Блок динамічної байєсівської мережі складається з блоку попередньої обробки даних, блоку побудови динамічної мережі Байєса та блоку розпізнавання символів.

### 8.3. Емпіричне оцінювання алгоритмів розпізнавання

#### 1. Опис експериментальних даних

Для проведення експерименту використано базу даних Google Patent Search. У цій базі даних представлені зареєстровані в США копії патентів, що були від скановані. Для роботи обрано перші сторінки з 10 патентів. З цих сторінок були

штучно виділені зображення маленьких літер англійського алфавіту. Всього використано 700 зображень літер. Кількість зображень для кожної літери відрізняється залежно від частоти появи літери в тексті. Наприклад, для символу ‘e’ кількість зображень є найбільшою – 86 зображень. Для символу ‘a’ використано 55 зображень.

Вхідну вибірку із 700 зображень розбито на навчальну та перевірочну наступним чином. В якості навчальної вибірки обиралися від 100 до 500 зображень літер. В якості перевірочної вибірки обиралося від 100 до 250 зображень. Вибірki не перетинаються. Спочатку випадково обиралося одне зображення навчальної вибірки для кожного символу. Після цього випадково обиралося решта зображень для навчальної та тестової вибірок.

Приклад зображень для символів наведено на рис. 8.5 – 8.6.



Рис 8.5. Приклад зображень для символу ‘e’.



Рис 8.5. Приклад зображень для символу ‘s’.

## 2. Оцінювання якості розпізнавання за допомогою найвного байєсівського класифікатора

### 1. Залежність точності розпізнавання від обсягу навчальної вибірки

Розпізнавання за допомогою найвного байєсового класифікатора. Зменшення зображень до розміру 10x10. Обсяг навчальної вибірки обирався від 100 до 600 зображень літер. 100 зображень використовувалося для перевірки якості роботи системи. Результати виконаних обчислювальних експериментів представлені в табл. 8.1.

Таблиця 8.1

Результати розпізнавання за допомогою найвного байєсівського класифікатора в залежності від розміру навчальної вибірки.

Розмір навчальної вибірки	Час навчання (мс)	Час тестування (мс)	Точність розпізнавання (мс)
100	374	374	37
200	734	374	48
300	1139	360	70
400	1564	371	75
500	1921	367	78
600	2312	359	78

На рис. 8.6 наведено графік залежності точності розпізнавання від обсягу навчальної вибірки.

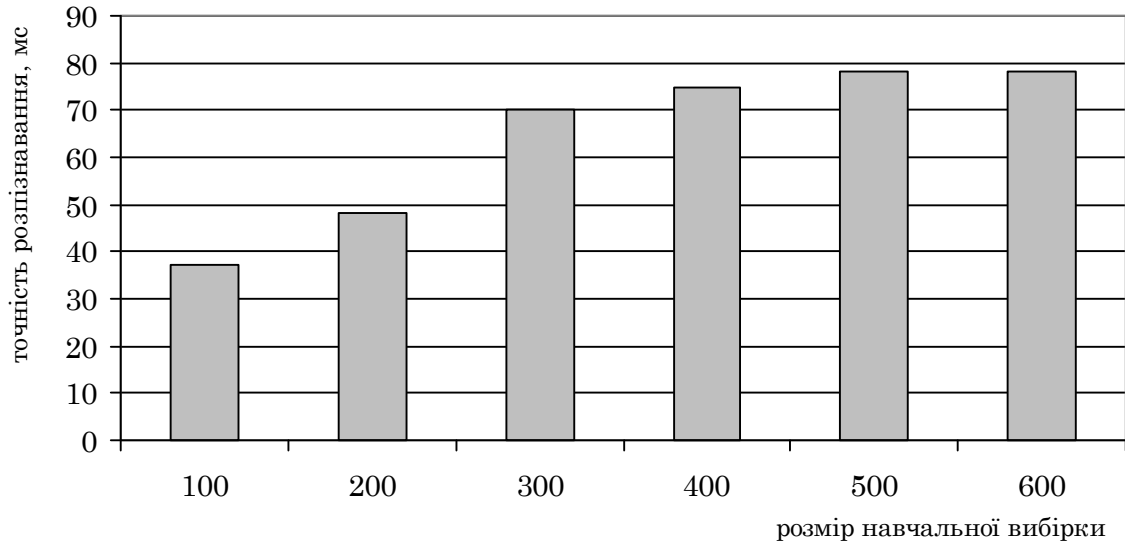


Рис 8.6. Гістограма точності розпізнавання за алгоритмом наївного байєсівського класифікатора залежно від розміру навчальної вибірки

*2. Точність розпізнавання залежно від розміру зображення*

Розпізнавання за допомогою наївного байєсового класифікатора. Обсяг навчальної вибірки складав 500 зображень літер, 200 зображень використовувалося для перевірки якості роботи системи. Вхідне зображення зменшувалося до розміру 8x8, 9x9, 10x10, 11x11, 12x12 або 15x15. Результати виконаних експериментів представлені в таблиці 8.2.

*Таблиця 8.2*

Результати розпізнавання за допомогою НБК для різних значень розміру зображення.

Розмір зображення	Час навчання (мс)	Час тестування (мс)	Точність розпізнавання (мс)
8	1094	442	74,5
9	1749	656	76
10	1921	703	78,5
11	2109	842	74
12	2360	952	72
13	3312	1359	71

На рис. 8.7 показано гістограму залежності точності розпізнавання для НБК від розміру зображення, а на рис. 8.8 – залежність часу навчання та тестування від розміру зображення.

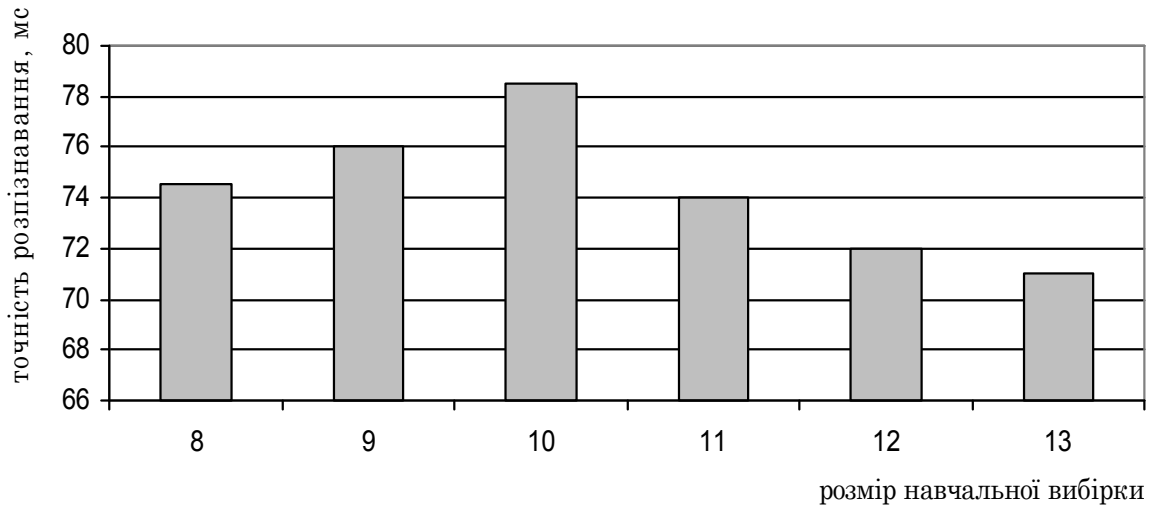


Рис 8.7. Гістограма точності розпізнавання для наївного байєсівського класифікатора в залежності від розміру зображення

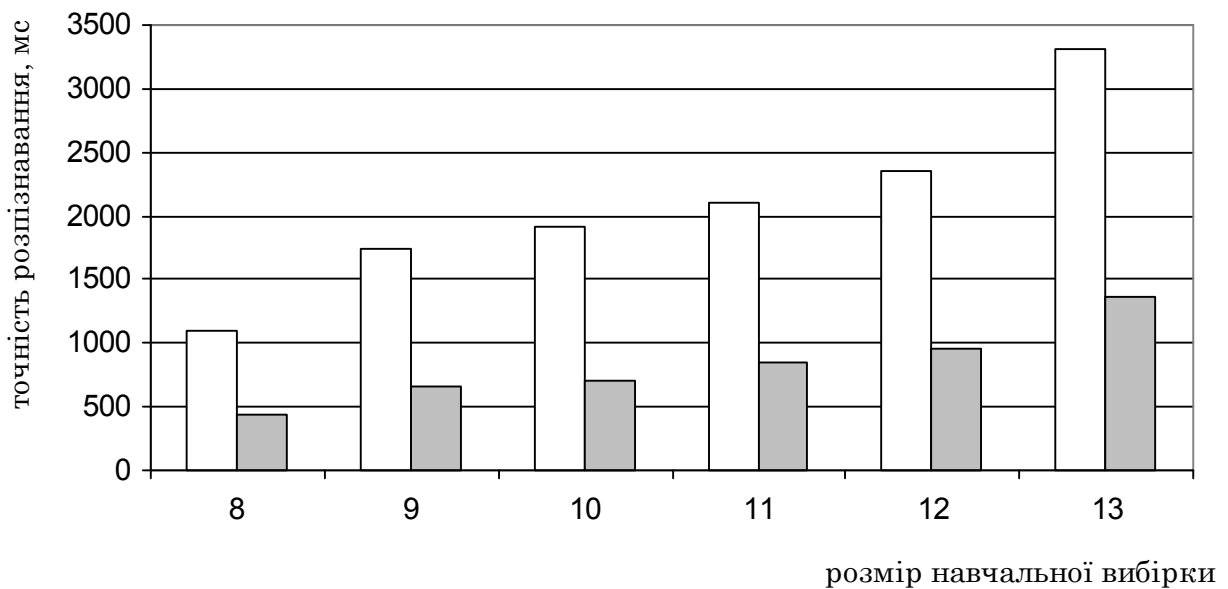


Рис. 8.8. Гістограми залежності часу навчання та тестування для наївного байєсівського класифікатора в залежності від розміру зображення

**Оцінювання якості розпізнавання за допомогою мережі Байєса зворотного висновку**

*1. Залежність точності розпізнавання від обсягу навчальної вибірки*

Розпізнавання за допомогою наївної байєсівської мережі з формуванням зворотного імовірнісного висновку. Зменшення зображень до розміру 23x23. Обсяг навчальної вибірки обирався від 100 до 600 зображень літер. 100 зображень використовувалося для перевірки якості роботи системи. Результати наведені в табл. 8.3.

Таблиця 8.3

Результати для мережі Байєса зворотного висновку залежно від параметрів

Розмір навчальної вибірки	Час навчання (мс)	Час тестування (мс)	Точність розпізнавання (мс)
100	1077	10000	82
200	1796	15281	84
300	3032	34031	89
400	4515	56218	89
500	5765	75062	90
600	6656	89921	90

На рис. 8.9 наведено гістограму залежності точності розпізнавання від обсягу навчальної вибірки.

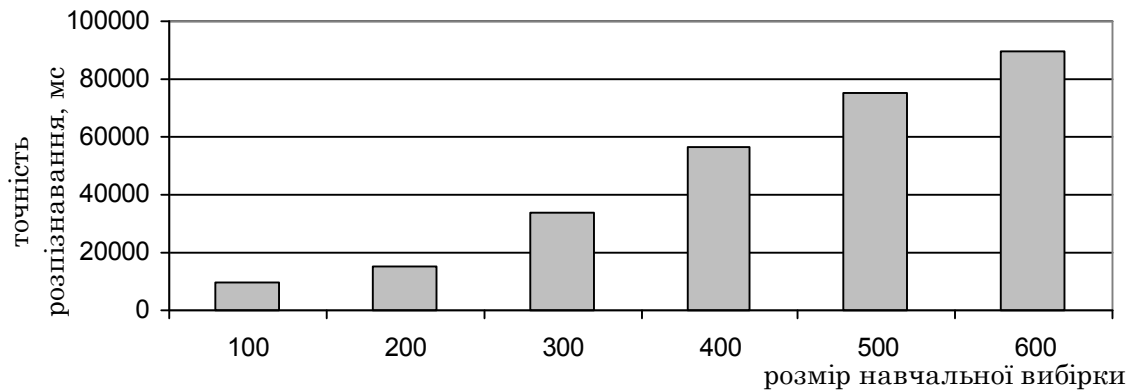


Рис 8.9. Гістограма точності розпізнавання для мережі Байєса зворотного висновку в залежності від обсягу навчальної вибірки

*2. Точність розпізнавання в залежності від розміру зображення*

Розпізнавання за допомогою наївної байєсівської мережі з формуванням зворотного імовірнісного висновку. Обсяг навчальної вибірки складав 450 зображень літер. 250 зображень використовувалося для перевірки якості роботи системи. Вхідне зображення зменшувалося до розміру 8x8, 10x10, 15x15, 20x20, 23x23, 25x25 або 30x30.

Таблиця 8.4

Результати розпізнавання за допомогою мережі Байєса з формуванням зворотного імовірнісного висновку залежно від розміру зображення

Розмір зображення	Час навчання (мс)	Час тестування (мс)	Точність розпізнавання (мс)
8	1077	10000	65,2
10	1796	15281	72,8
15	3032	34031	83,8
20	4515	56218	88,4
23	5765	75062	89,6
25	6656	89921	89,2
30	7750	148186	87,3

На рис. 8.10 показана гістограма залежності точності розпізнавання для наївного байєсівського класифікатора від розміру зображення, а на рис. 8.11 – залежність часу навчання та тестування від розміру зображення.

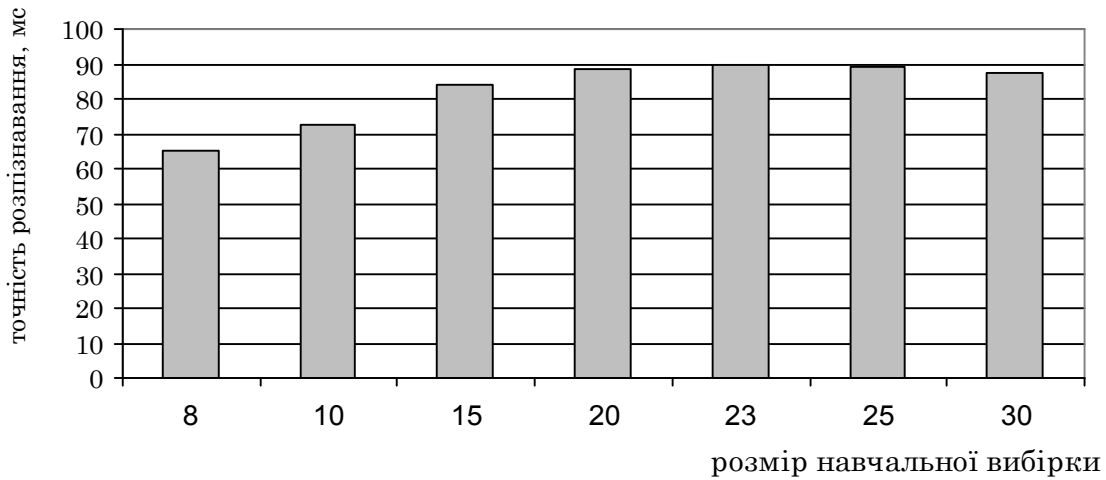


Рис 8.10. Гістограма точності розпізнавання для мережі Байєса зворотного висновку в залежності від розміру зображення

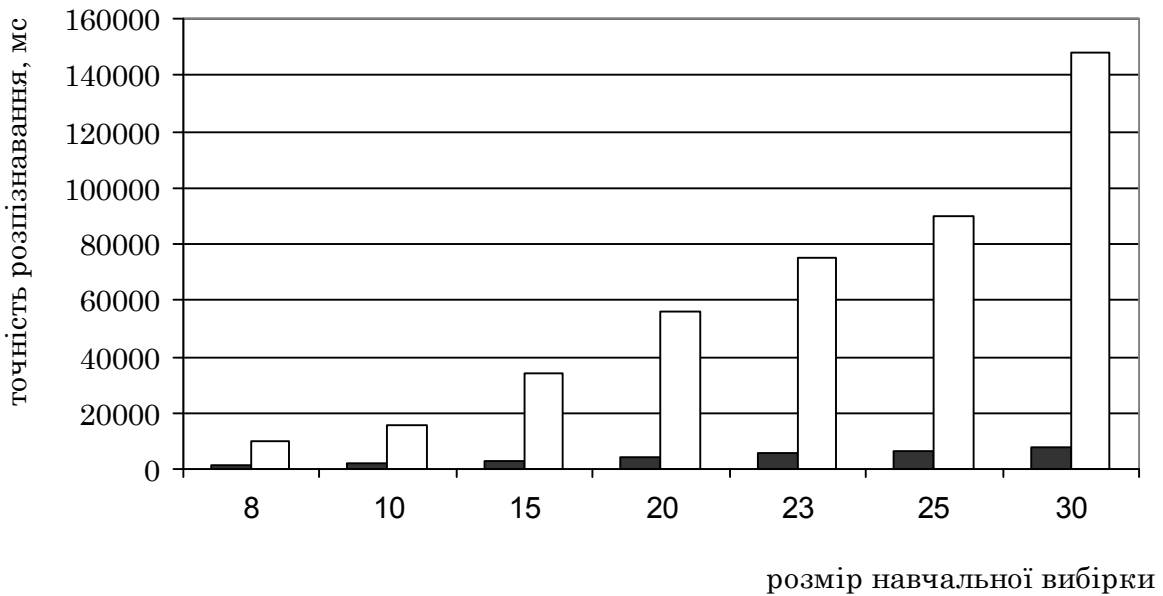


Рис.8.11. Час навчання та тестування для мережі Байєса зворотного висновку в залежності від розміру зображення

**Оцінка якості розпізнавання за допомогою динамічної мережі Байєса**

*1. Залежність точності розпізнавання від кількості прихованих станів*

Розпізнавання за допомогою динамічної байєсівської мережі. Розмір навчальної вибірки – 5600 зображень. Розмір перевірконої вибірки – 200 зображень. Розмір зображень становив 20x20.

Таблиця 8.5

Результати розпізнавання для динамічної мережі Байєса в залежності від кількості прихованих станів

Кількість прихованих станів	Час навчання, мс	Час тестування, мс	Точність розпізнавання, мс
4	49781	65375	81
6	166858	91329	87
8	286016	114098	91
10	368516	161031	90
12	392244	177496	87

На рис. 8.12 наведена гістограма залежності точності розпізнавання від кількості прихованих станів, а на рис. 8.13 – час навчання та тестування в залежності від кількості прихованих станів.

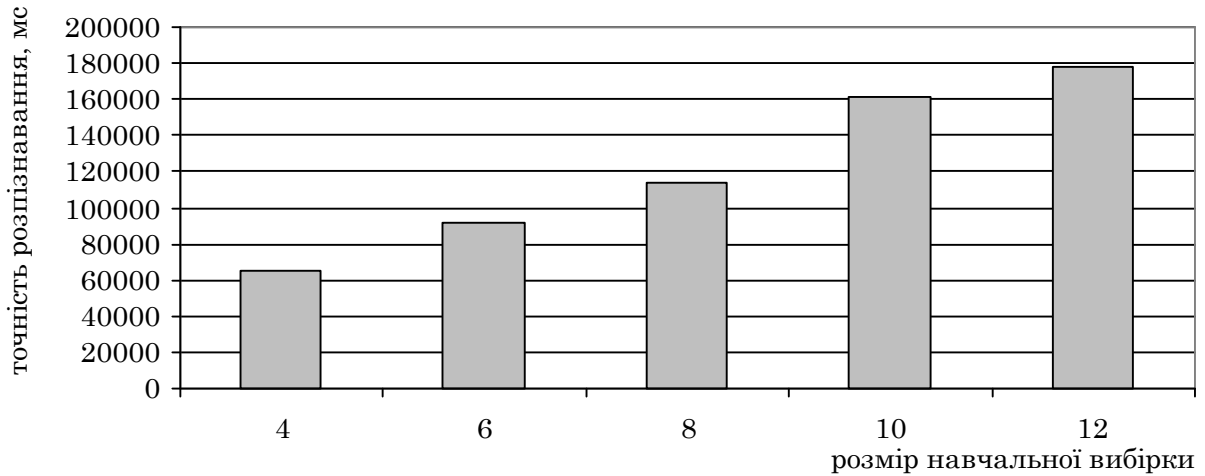


Рис 8.12. Гістограма точності розпізнавання динамічної байєсівської мережі в залежності від кількості прихованих станів

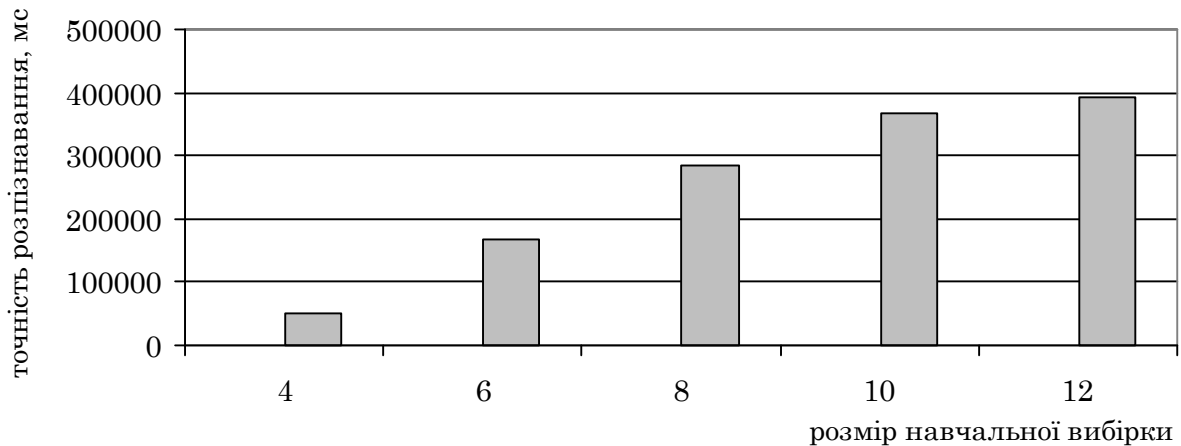


Рис. 8.13. Гістограма часу навчання та тестування для динамічної мережі Байєса в залежності від кількості прихованих станів

*2. Точність розпізнавання в залежності від розміру зображення*

Розпізнавання за допомогою динамічної байєсівської мережі. Розмір зображення: 20x20. Використовувалося 8 прихованих станів, що є оптимальною кількістю прихованих станів.

Обсяг навчальної вибірки обирався від 100 до 600 зображень літер. 100 зображень використовувалося для перевірки якості роботи системи. Результати проведених експериментів представлені в табл. 8.6.

*Таблиця 8.6*

Залежність результатів розпізнавання динамічної мережі Байєса від навчальної вибірки

Розмір навчальної вибірки	Час навчання, мс	Час тестування, мс	Точність розпізнавання, мс
100	172824	117656	72
200	172064	116265	75
400	175829	110983	91
600	176829	114874	91

На рис. 8.14 наведено гістограму залежності точності розпізнавання від обсягу навчальної вибірки. Як видно з графіка, точність розпізнавання збільшується із зростанням розміру вибірки для тестування.

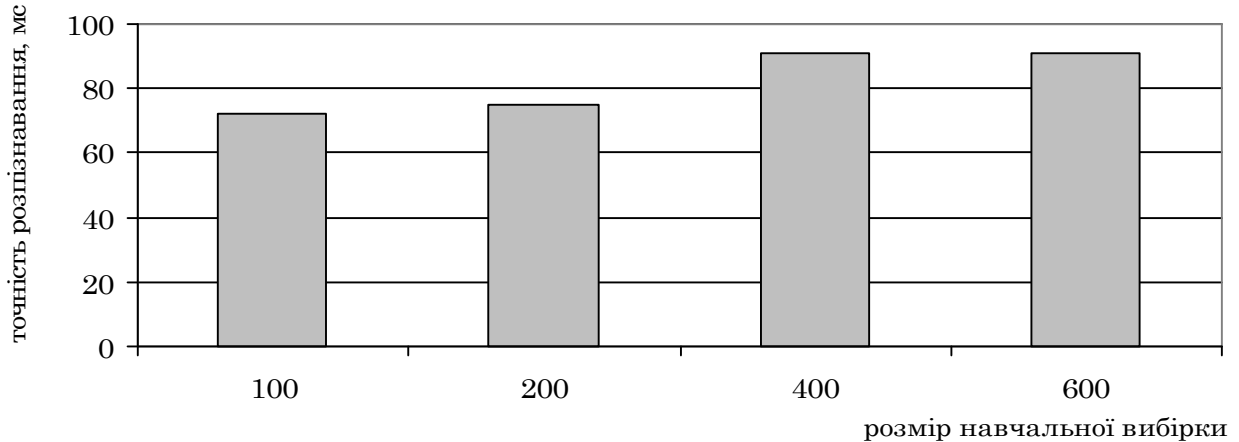


Рис. 8.14. Гістограма залежності точності розпізнавання від розміру навчальної вибірки для динамічної мережі Байєса

***Аналіз результатів та порівняння результатів, одержаних різними методами***

Виконані обчислювальні експерименти на вибірці із 700 зображень, що були отримані з патентної бази даних Google Patent Search, показали, що при збільшенні обсягу навчальної вибірки точність розпізнавання підвищувалася. Разом з цим також підвищуються час навчання та розпізнавання. Наївний байєсівський класифікатор досягає найкращої точності при заданих параметрах при розмірі навчальної вибірки від 500 зображень (79%). У той же час, як мережа Байєса із



зворотним формуванням імовірнісного висновку досягає більшої точності вже при 100 навчальних зображеннях (82%), що є найкращим результатом для такого розміру навчальної вибірки.

Для порівняння результатів розпізнавання з іншими методами було виконано розпізнавання окремих символів за допомогою ПП "ABBY FineReader 7".

Програма розпізнавала всі 700 зображень окремих літер. Правильно було розпізнано 652 зображення літери. 48 зображень літер було розпізнано помилково, тобто похибка розпізнавання склала 7%.

У табл. 8.7. наведено приклади помилок при розпізнаванні за допомогою реалізованих методів та за допомогою програми «FineReader».

Таблиця 8.7

Приклади помилок при розпізнаванні

Зображення	Прокорильне значення зображення	Результат розпізнавання			
		наївний байєсівський класифікатор	мережа Байєса зворотного висновку	динамічна мережа Байєса	Fine Reader
	t	r	t	t	t
	d	a	d	d	J
	b	a	h	b	b
	t	t	t	l	t

Результати розпізнавання для кожного з реалізованих методів, а також за допомогою програми FineReader наведені в табл.8.8.

Таблиця 8.8

Порівняння результатів розпізнавання

Назва методу	Похибка розпізнавання (%)	Середній час розпізнавання одного символу (мс)
Наївний байєсівський класифікатор	21	4,5
Наївна мережа Байєса зі зворотним формуванням імовірнісного висновку	10	300
Динамічна мережа Байєса	9	1340
FineReader	7	-

Найменша похибка при розпізнавання була отримана програмою FineReader, вона склала 7%. Серед інших методів найкращу точність дала динамічна мережа Байєса, але цей метод вимагає багато часу для розпізнавання одного символу, який перевищує 1 с, що є небажаним при використанні методу на практиці. Наївна мережа Байєса із зворотним формуванням імовірнісного висновку також показує хороший результат, похибка склала 10%. Найгіршим виявився наївний байєсівський класифікатор, для якого похибка розпізнавання склала 21%.

#### **8.4. Огляд комп'ютерних систем для розпізнавання символів**

##### ***Інтелектуальна система розпізнавання документів «ABBYY FineReader»***

Остання версія цієї системи: ABBYY FineReader 11. Нова версія добре зображення низької якості: факси, знімки, зроблені за допомогою фотокамери мобільного телефону. У програмі поліпшені механізми визначення структури документа, в тому числі: знаходження таблиць, картинок, графіків.

Основні можливості системи такі:

1. Збереження повного оформлення.

Використовується технологія адаптивного розпізнавання ADRT®2.0 (Adaptive Document Recognition Technology). Документ аналізується як єдине ціле, при цьому повністю зберігається його логічна структура.

2. Інтерфейс. Інтерфейс програми містить вікно з типовими сценаріями. Для зручності використання сценарії поділені на групи, налагодження збереження винесені у це ж вікно. Користувач вибирає потрібний сценарій конвертування і за одне клацання "мишки" одержує копію документа, що редагується.

3. Сканування. Нова версія програми підтримує роботу з усіма популярними моделями сканерів і багатофункціональних обладнань, а також коректно розпізнає фотографії текстів, зроблені за допомогою цифрового фотоапарата або фотокамери мобільного телефону. Для обробки зображень, зроблених за допомогою фотокамери мобільного телефону, у програмі передбачено такі функції: виправлення трапецієподібних викривлень, усунення шумів і нечіткості зображення.

4. Підтримка мов. ABBYY Finereader розпізнає документи на 186 мовах на основі кирилиці, латиниці, грецького й вірменського алфавітів, підтримуються мови на основі ієрогліфічного письма. У нову версію програми додана нова мова розпізнавання – корейська, яка продовжує розширювати перелік неалфавітних мов. Для 39 мов передбачена перевірка орфографії.

5. Автоматичне визначення мови розпізнавання. Використовуючи інтелектуальні алгоритми, ABBYY Finereader 11 автоматично одержує інформацію про мови, на яких написаний документ, і використовує її при розпізнаванні.

6. Оптимізація для багатоядерних процесорів.

У програмне забезпечення системи OCR Finereader включаються словники для допомоги алгоритмам розпізнавання. Словники надають довідки у багатьох випадках, але швидко відмовляють, коли, наприклад, мають справу із власними іменами, які не знаходяться в словнику. Цей ефект особливо помітний, оскільки програма частіше ніж в середньому по всіх символах, помиляється у словах, які відсутні в її словнику.

##### ***Система оптичного розпізнавання документів "OCR CuneiForm"***

OCR CuneiForm – це система оптичного розпізнавання російського розробника програмного забезпечення Cognitive Technologies.

Основні можливості OCR CuneiForm:

1. Інтерфейс. Інтерфейс програми містить контекстні меню, що випадають, панелі швидкого доступу, контекстну допомогу.

2. Сканування.

а) Автоматичний підбор оптимальних параметрів сканування.

б) Можливість імпортування сканованих або отриманих через факс- модем графічних файлів у багатьох форматах.

в) Обробка зображень: друк, інвертування, поворот.

3. Фрагментація.

а) Автоматичний і напівавтоматичний режими пошуку блоків тексту, таблиць і графіки, який забезпечує більшу гнучкість при роботі із багатоколонковими текстами, текстами складної структури і з графічними елементами.

б) Режим ручної фрагментації для роботи з текстами особливо складної структури.

4. Розпізнавання. Cuneiform розпізнає будь-які поліграфічні, машинописні гарнітури всіх написань і шрифти, одержувані із принтерів за винятком декоративних і рукописних. У систему вбудовані спеціальні алгоритми для розпізнавання тексту з матричного принтера, друкованої машинки, поганих ксерокопій і факсів. Система побудована на адаптивних алгоритмах із самонавчанням, що сприяє підвищенню ймовірності отримання правильного результату.

5. Підтримка мов. Система розпізнає російську, англійську, змішану російсько-англійську, українську, німецьку, французьку, іспанську, португальську, італійську, голландську, данську, шведську, фінську, сербську, хорватську, польську, казахську, узбецьку та інші мови.

6. Словниковий контроль. Словник загальноживаної лексики кожної підтримуваної мови для контекстної перевірки й підвищення якості результатів розпізнавання. Можливість створення і поповнення користувацького словника, а також можливість експорту/імпорту словника текстових файлів.

7. Реалізація принципу "What You Scan Is What You Get". Cuneiform дозволяє отримати повну копію документа, що вводиться, включно із:

а) Шрифтовим оформленням і форматуванням.

б) Розташуванням тексту, ілюстрацій і таблиць.

в) Стовпчиками, абзацами, відступами, стилями і розмірами шрифтів.

г) Чорно-білими, 256-градаційними сірими і кольоровими 24-бітовими ілюстраціями у вихідному RTF-файлі.

8. Забезпеченням роботи з таблицями.

9. Редагуванням. У програму вбудовано багатфункціональний редактор, що не поступається за своїми можливостями популярним текстовим процесорам.

а) Забезпечується одночасне підсвічування розпізнаного тексту і вихідного зображення; підтримується функцією "наступний/попередній сумнівно розпізнаний".

б) Підтримка ілюстрацій, таблиць, колонок, колонтитулів, складного форматування й різних шрифтів.

в) Існує можливість редагування текстових документів популярних форматів.

10. Інтегрування з іншими прикладними програмами Опції командного рядка і підтримка Drag&drop для виклику із зовнішніх програм, сканування, розпізнавання і збереження результатів в автоматичному режимі.

### ***On-line сервіс розпізнавання тексту "Onlineocr.ru"***

Сервіс дозволяє розпізнавати текст або символи з будь-якого зображення (сканований документ, цифрова фотографія або просто картинка у форматі JPG,

BMP, TIF і ін.). Onlineocr.ru обробляє також багатосторінкові документи TIFFF і PDF.

Отриманий у результаті розпізнавання текст конвертується в такі вихідні формати: Adobe PDF, MSWord, Excel, Html, Rtf, Txt. Створений файл буде повторювати структуру вихідного документа (таблиці, стовпчики, шрифти і т. ін.).

Сервіс дозволяє зберігати результати у віртуальному робочому кабінеті online, на жорсткий диск, редагувати, відправляти поштою й роздруковувати на принтері. Загалом, основною перевагою сервісу є простота використання, зокрема відсутність необхідності встановлювати програмний продукт. Недоліками є порівняно низька функціональність та якість розпізнавання.

### ***Комп'ютерна програма «Caere OmniPage Professional»***

Остання версія – Caere OmniPage Pro 18. Програма використовує алгоритм, заснований на знаходженні загальних специфічних особливостей символів.

Ця система містить 100 різних алгоритмів для ідентифікації 100 різних символів: верхнього й нижнього регістру від "A" до "Z", записи чисел і символів пунктуації. Кожний із цих алгоритмів шукає "особливості" написань типу "островів", "півостровів", крапок, прямих відбитків і дуг. Експертні системи також розглядають горизонтальні й вертикальні проекції відбитків букви й звертають увагу на основні особливості в створених кривих, підсумовуючи у них кількість темних пікселів.

### ***Numenta Platform for Intelligent Computing***

Numenta Platform for Intelligent Computing (NuPIC) – це перша реалізація технології НТМ компанії Numenta. Платформа дозволяє будувати мережі НТМ.

Платформа NuPIC складається з двох компонентів: Numenta Tools Framework та Numenta Runtime Engine. Пакет NuPIC містить набір прикладів коду та інструкцію програміста.

Побудова НТМ мережі.

1. Визначення задачі. Наприклад, якщо НТМ мережа була навчена розпізнавати вигулювання собак певним шляхом, то в подальшому вона зможе взнати собаку з різних кутів і відстаней. Якщо НТМ навчається на зображеннях різних собак, вона вчиться класифікувати нові собаки, як собаки.

2. Збір та представлення даних. Необхідно мати достатню кількість даних для побудови моделі. Дані повинні бути представлені у вигляді, який Numenta Runtime Engine зможе сприйняти.

3. Розробка та налаштування НТМ мережі. Використовуючи Python API, який включено до Numenta Tools Framework, розробник створює ієрархію вузлів, які будуть оброблювати дані. Програма інстанціює та пов'язує вузли мережі НТМ, налаштовує параметри. Результатом цього процесу є набір сценаріїв, які створюють НТМ мережі, пов'язані з ними вузли і параметри, а потім зберігають файл НТМ мережі.

4. Запуск НТМ мережі для навчання та виводу. Розробник використовує Python Session API для створення сценарію, який виконується за допомогою Numenta Runtime Engine. Оскільки навчання займає деякий час, сценарій зазвичай зберігає файл НТМ мережі декілька разів у ході навчання. Навчена мережа в подальшому може бути завантажена в Numenta Runtime Engine для обробки нових даних.

5. Пошук і усунення несправностей, аналіз і поліпшення мережі НТМ.

Після виконання розробником всіх кроків побудови НТМ мережі, часто ітераційним чином, мережа є готовою для використання.

Програма є потужною системою для розпізнавання, але вимагає досить високих знань для її застосування, що робить неможливим її широке використання.

### ***Numenta Vision Toolkit***

Програма Vision Toolkit (Beta) дозволяє створювати власні системи розпізнавання зображень з використанням технології НТМ.

Використання Vision Toolkit включає такі кроки:

Крок 1. Збір зображень для навчання. Необхідно завантажити велику кількість зображень для кожного класу. Виконати збір даних можна використовуючи інструмент веб-пошуку.

Крок 2. Навчання. Програма виконує автоматичне навчання мережі НТМ.

Крок 3. Розпізнавання нових зображень. Після імпорту нових зображень система розпізнає клас цих зображень.

Програма представляє собою зручну систему для розпізнавання зображень будь-якої природи. Недоліком є те, що система знаходиться на стадії розробки.

### ***Порівняльний аналіз програм***

Для порівняння виконано розпізнавання окремих символів за допомогою програми FineReader, яка показала найкращу точність розпізнавання. У той же час необхідно відзначити, що реалізовані мережі Байєса із зворотним формуванням імовірнісного висновку та динамічна мережа Байєса також показали прийнятні результати. У табл. 8.9 наведено порівняльний аналіз деяких з існуючих програмних рішень.

*Таблиця 8.9*

Порівняльний аналіз існуючих рішень

Назва програми	Вартість	Призначення	Переваги	Недоліки
OCR CuneiForm	Безкоштовна	Розпізнавання документів	Висока функціональність, безкоштовність	Невелика кількість підтримуваних мов
АВВУ FineReader 10	700 - 1400 грн.	Розпізнавання документів	Висока функціональність, висока точність розпізнавання, зручний інтерфейс	Висока вартість
onlineocr.ru	До 1 російського рубля за одну сторінку	Розпізнавання документів	Простота використання	Низька функціональність
Caere-OmniPage Pro 9.	\$499.99	Розпізнавання тексту	Висока точність розпізнавання	Висока вартість
NuPIC	Безкоштовна для дослідників	Побудова мереж НТМ для розпізнавання	Потужна система, безкоштовна	Необхідність високої підготовки спеціаліста
Numenta Vision Toolkit (Beta)	Комерційна	Розпізнавання зображень	Можливість розпізнавання зображень довільної природи	Знаходиться на стадії розробки

### Контрольні питання

1. Надайте опис та пояснення методики розпізнавання символів за допомогою наївної мережі Байєса зі зворотним формуванням імовірнісного висновку.
2. Що таке динамічна байєсівська мережа, її роль та місце серед методів розпізнавання? Наведіть приклади використання.
3. Опишіть алгоритм розпізнавання текстових символів на основі мереж Байєса.
4. Які три основні задачі при моделюванні моделей Маркова необхідно вирішити? Яким чином кожна з них вирішується?
5. Надайте обґрунтування алгоритму Баума-Велча.
6. Дайте опис алгоритму Баума-Велча для випадку неперервних спостережуваних величин.
7. Поясніть по кроках методику розпізнавання символів на основі динамічної мережі Байєса.
8. Дайте опис структури програмної реалізації системи розпізнавання символів на основі мереж Байєса.
9. Наведіть приклади можливих помилок при вирішенні задачі розпізнавання різними методами.
10. Які сучасні комп'ютерні системи для розпізнавання існують? Дайте опис та приклади використання.
11. Як оцінити якість розпізнавання символів за допомогою наївного байєсівського класифікатора?
12. Як залежить точність розпізнавання від обсягу навчальної вибірки?
13. Порівняйте якість результатів розпізнавання символів різними методами
14. Вкажіть переваги та недоліки системи оптичного розпізнавання документів "OCR CuneiForm"
15. Як залежить точність розпізнавання символів від розміру зображення?

## РОЗДІЛ 9 СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ МОДЕЛЮВАННІ ПРОЦЕСІВ РІЗНОЇ ПРИРОДИ ТА ЇХ ПРОГРАМНА РЕАЛІЗАЦІЯ

- Задачі, які успішно вирішуються із застосуванням байєсівських мереж
- Огляд комп'ютерних програм для створення та застосування мереж Байєса
- Приклади побудови систем підтримки прийняття рішень та застосування програмних продуктів для побудови мереж Байєса

### 9.1. Огляд задач, які успішно вирішуються із застосуванням байєсівських мереж

Застосування комп'ютерних інформаційних систем у процесі підтримки прийняття рішень стає невід'ємною частиною професійної діяльності фахівців бізнес-аналітиків та менеджерів вищої ланки, соціологів та науковців, а й пересічних підприємців, лікарів, інженерів тощо. Адже все частіше вони стикаються з необхідністю не лише адекватно реагування на поточну ситуацію, що часто характеризується наявністю невизначеностей, нечіткою та неструктурованою інформацією, а й передбачити можливий розвиток подій та їх наслідків в майбутньому. Розвиток сучасних інформаційних технологій надає користувачам широке коло потужних інструментів для прийняття оптимальних рішень, одним із яких є інтелектуальний аналіз даних. У рамках технології головна цінність інтелектуального аналізу даних їх полягає у здатності виявляти невідомі та нетривіальні зв'язки між факторами, про які іноді навіть експерти у відповідній предметній області можуть не мати достатньої уяви. Байєсівські мережі зарекомендували себе надійним та ефективним засобом, здатним виконувати майже всі задачі такого роду.

Саме тому мережі Байєса як засіб інтелектуального аналізу даних активно впроваджуються в багатьох сферах [14], і, зважаючи на стрімкість та різноплановість даного процесу, відзначити всі області їх застосування неможливо (табл. 9.1)[15].

Таблиця 9.1

Напрями використання мереж Байєса

Галузь	Напрями
Промисловість	Енергетика, оборона, безвідмовність, робототехніка, будівництво
Природничі та гуманітарні науки	Медицина, генетика, екологія, геологія, географія, судмедекспертиза
Послуги	Банківська справа, фінанси, маркетинг, законодавство
Інформаційні технології	Моделювання, дослідження операцій, промислові інформаційні системи, інформаційний пошук

### *Медицина та біологія*

Забезпечення якісної та оперативної медичної діагностики залишається складною задачею, оскільки характеризується наявністю невизначеностей, відсутністю повної та об'єктивної інформації щодо стану пацієнта, необхідністю опрацювання у стислі терміни значних обсягів довідкової інформації, необхідної для встановлення діагнозу та призначення лікування.

Накопичення медичних знань щодо засобів та методів лікування, широкий спектр діагностичних тестів потребують їх відповідної обробки для підтримки прийняття рішень фахівцями-медиками. Особливо це важливо на початковій стадії лікування, коли ставиться попередній діагноз і визначаються можливі напрями лікування пацієнта. Крім того, зростання вартості лікування і ризику судових позовів з боку невдоволених пацієнтів спричинює тиск на медичну сферу і спонукає шукати шляхи зниження вартості лікування та зменшення кількості лікарських помилок. Все це сприяє активному впровадженню медичних систем підтримки прийняття рішень. Розвиток інформаційних технологій дозволив об'єднати накопичені медичні знання та історії хвороб пацієнтів у централізовані бази даних та знань, створити значну кількість різноманітних експертних систем. Медицина стала однією з перших сфер, де мережі Байєса знайшли своє застосування. Системи на їх основі орієнтовані на встановлення діагнозу, прогнозування результатів різних методів лікування, вибору оптимального індивідуального курсу лікування для пацієнта [34].

Застосування мереж Байєса при постановці медичних діагнозів розпочалось близько двох десятиріч тому. Першою експертною системою на основі мереж Байєса була CONVINCENCE [183], створенню якої передувала розроблена у 1980 році в Пітсбургському університеті швидка медична довідка – QMR-система (Quick Medical Reference). CONVINCENCE складається із статистичних та експертних даних, які являють собою один із діагностичних інструментів лікаря-терапевта.

Однією з перших медичних діагностичних систем є PATHFINDER, розроблена для діагностики захворювань лімфовузлів [16, 184], що реалізує 60 різноманітних варіантів постановки діагнозу та має 130 змінних-симптомів, значення яких можуть спостерігатися при вивченні клінічних випадків. Якість діагностики, забезпечувана системою наблизилася до рівня експертів, що сприяло зростанню її популярності (програмне забезпечення PATHFINDER починаючи з четвертої версії є платним). Слід відзначити також інші медичні експертні системи, зокрема NESTOR – систему діагностування ендокринологічних порушень [185], MUNIN – систему діагностування нервомускулярних порушень [186], PATHFINDER IV – систему діагностування захворювань лімфатичної системи [187], QMR (QuickMedicalReference) – діагностичну систему лікаря-терапевта [188], DIAVAL – експертну систему для ехокардіографії [189], а також розробки компанії Knowledge Industries, Inc. (KI), які охоплюють діагностування та лікування порушень сну, психічні патології, травматологію, аналіз стану руки і зап'ястя, дерматологію та оцінку здоров'я в домашніх умовах, Hepar II – систему діагностування захворювань печінки [191], а також системи Child, Munin, Painulium, SWAN [192].

Можливості застосування мереж Байєса в медицині представлені на демонстраційному сайті [www.symptomedix.com](http://www.symptomedix.com) на прикладі інтерактивної діагностики головного болю.



В якості загального прикладу застосування мереж Байєса у медичній діагностиці, можна навести діагностування захворювань печінки за допомогою системи Нерар II, створеної для зменшення кількості процедур біопсії печінки [149]. Система Нерар II сконструйована за гібридним підходом: структура мережі побудована на основі знань експертів і відповідній медичній літературі, а числові дані для моделі, тобто ймовірнісний розподіл для вершин – з бази даних проб печінки Нерар, яка була створена у 1990. На момент створення системи Нерар II у базі даних було 699 записів. Нині ця база даних містить більше 800 записів з історій хвороб пацієнтів і постійно доповнюється. Варто зазначити, що ні один із записів у цій базі даних не є повним, а на момент створення бракувало 7792 значень (15,9% від усіх даних). Мережа Байєса системи Нерар II має близько 70 вершин, 11 з них – це різні хвороби печінки, а 61 – дані, що надав пацієнт, ознаки, симптоми хвороб і результати лабораторних тестів. Загалом система містить 1488 числових параметрів. На рис. 9.1 наведено спрощений фрагмент мережі Байєса системи Нерар II.

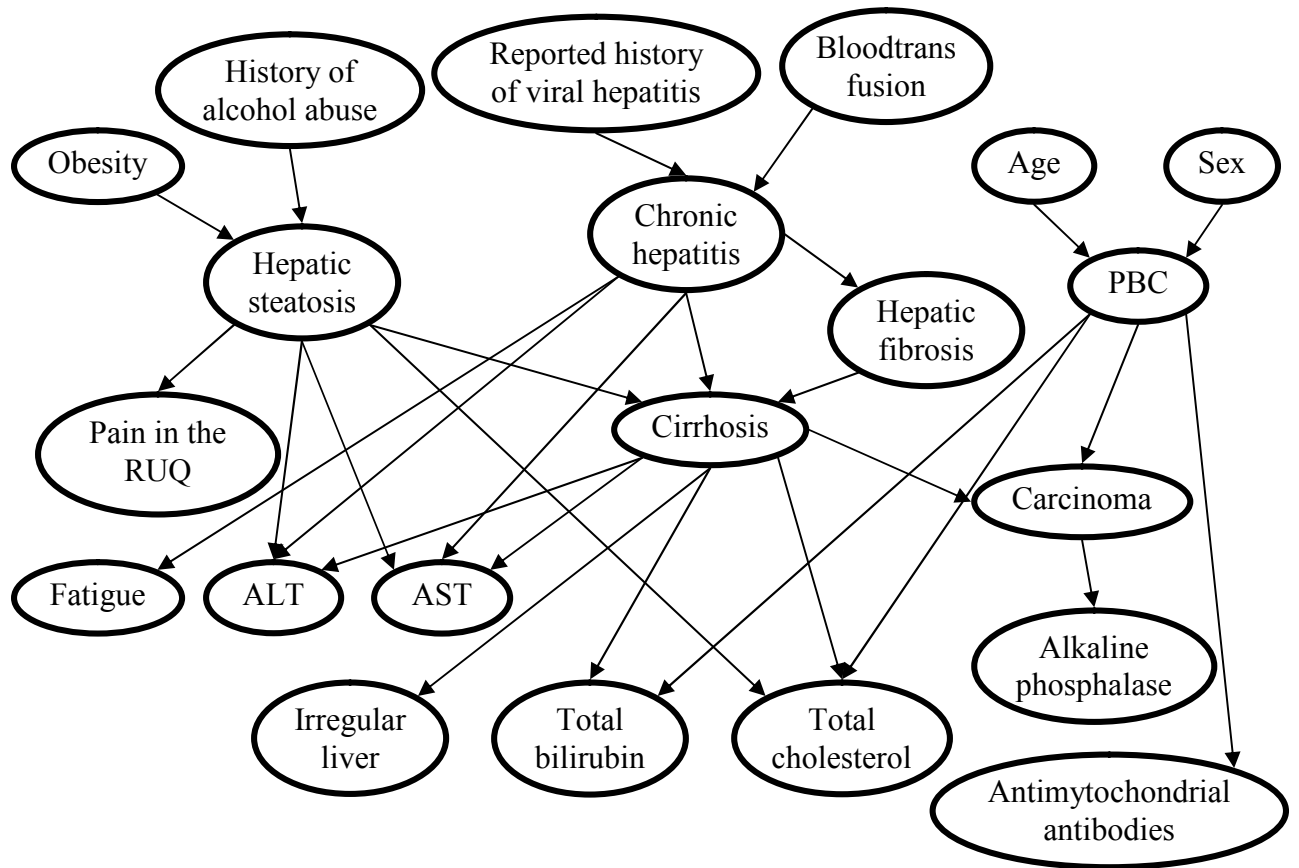


Рис. 9.1. Фрагмент мережі системи Нерар II

Для побудови ймовірнісного висновку в Нерар II використовувались засоби таких програмних пакетів для роботи з байєсівськими мережами, як SMILE та GeNIe. Дослідження ефективності роботи системи відбувалось за участі 23 медичних фахівців. Першим результатом було те, що в діагнозах 10-ти пацієнтів, випадковим чином вибраних з бази даних, діагнози, поставлені за допомогою НерарII виявились точнішими ніж поставлені фахівцями-медиками 70 % точних діагнозів порівняно з 33,1 %. Іншим важливим результатом дослідження виявилось те, що правильний висновок системи спонукав лікаря скоригувати своє рішення і таким чином медики підняли відсоток точності своїх діагнозів з 33,1 до 65,8 %.

У 2001 році було виконано порівняння Нерар II з Нерар-RB – системою для діагностування захворювань печінки на множині правил, яке показало, що обидва підходи є ефективними, хоча і мають свої переваги та недоліки [150]. Підхід на базі правил дає можливість протестувати модель, послідовно ідучи по шляху, за яким система приймає рішення, що можливо і для мереж Байєса, але автоматичне генерування пояснень в них не реалізовано. Важливою особливістю мереж Байєса є можливість отримання структури моделі з бази даних. Системи ж на базі правил краще застосовувати у випадках, коли причина-наслідок – не основний принцип, закладений у задачу або ж коли вона занадто складна, щоб бути представленою каузальною мережею. Крім того, як підтвердили експерименти, система на основі множини правил має труднощі з обробкою неповних даних – в системі Нерар-RB близько 35% пацієнтів не отримали діагнозу, тоді як в Нерар II ця цифра становила лише 2%.

Іншим прикладом медичної діагностичної системи є система PATHFINDER, розроблена Хекерманом та його колегами. Система PATHFINDER призначена для діагностики захворювань лімфовузлів [16]. На рис. 9.2 зображено фрагмент мережі Байєса для цієї системи.

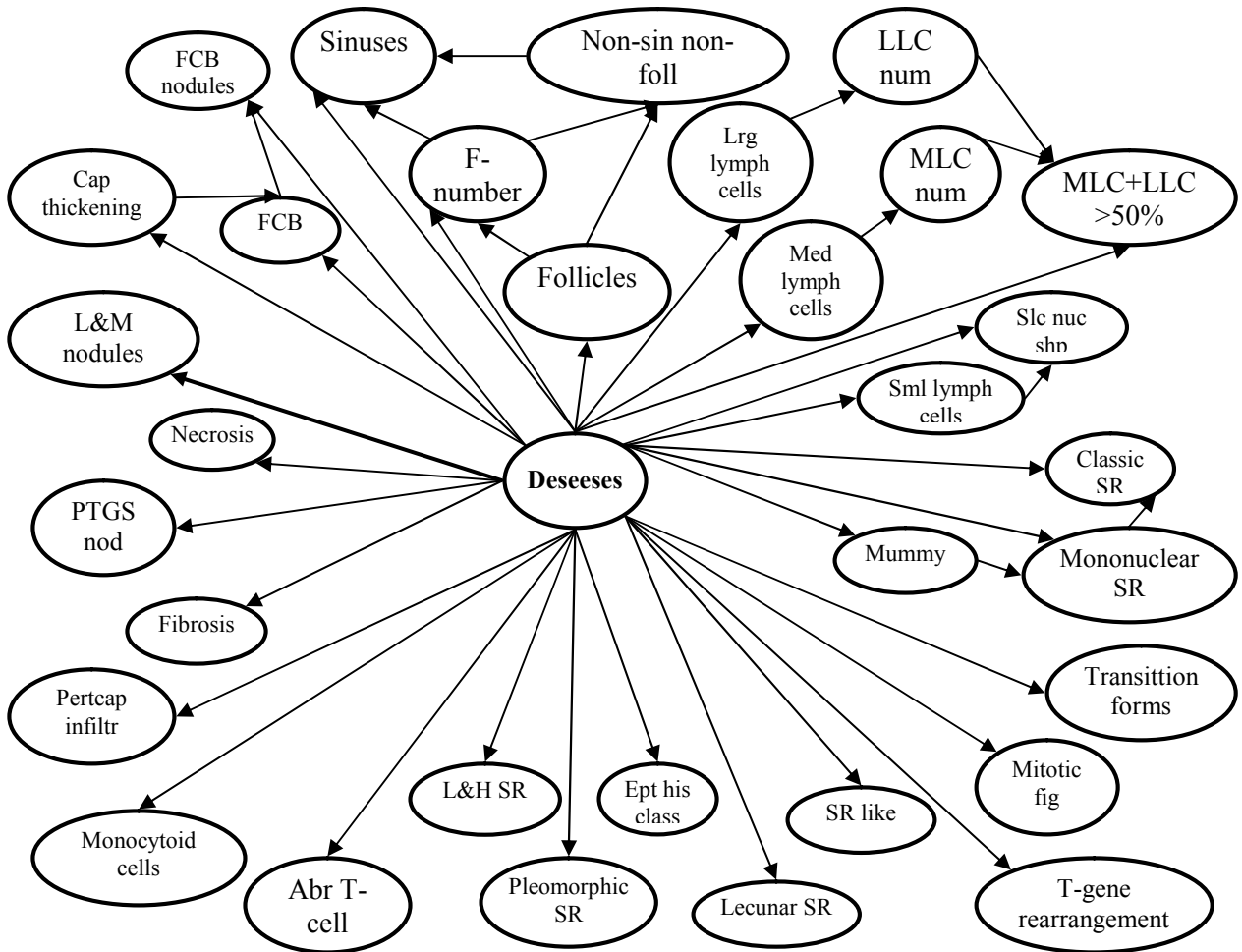


Рис. 9.2. Фрагмент системи PATHFINDER

PATHFINDER включає 60 різноманітних варіантів діагнозу та 130 змінних-симптомів, значення яких можуть спостерігатися при вивченні клінічних випадків. Система змогла наблизитися до рівня експертів, і її четверта версія набула статусу комерційної системи – Intellipath.

Для медичного прогнозування стану пацієнта за наявними симптомами та підбору методів лікування на основі мереж Байєса розроблено значно менше систем, ніж для діагностування. Серед них варто виділити систему прогнозування виживання пацієнтів із злоякісною шкірною меланомою [193] і систему NasoNet, призначену для прогнозування раку носоглотки [194] та систему MENTOR, призначену для прогнозування олігофренії у немовлят [195].

Прикладом застосування мереж Байєса в біології є програмна система SEMPNU, розроблена Фрідманом та ін. В ній, на основі методології вивчення причинних взаємозв'язків між генами, аналізі даних стосовно експресії генів [196], методі реконструкції філогенетичного дерева та на основі принципу максимальної правдоподібності, реалізовано філогенетичну реконструкцію [197].

### ***Військове застосування, техніка, безпека***

Мережі Байєса знайшли своє застосування і у військовій справі. Дослідницькою лабораторією ВМФ Австралії ще у кінці 90-х років минулого століття була розроблена навчальна модель “Operation Dardanelles”, призначена для моделювання тактичної задачі охорони територіальної зони з моря [200]. Ця модель включає в себе різні тактичні сценарії поведінки конфліктуючих сторін, дані про переміщення суден, розвідки та інші змінні. Послідовне надходження інформації про дії супротивників дозволяє синхронно прогнозувати ймовірності розвитку подій протягом всього конфлікту [201].

Необхідність ефективних засобів боротьби з тероризмом в сучасному світі ні в кого не викликає сумнівів. Антитерор є захисною позицією, спрямованою проти терористичних загроз. Він включає в себе збільшення обізнаності про потенційні загрози, стримування агресорів, розвиток захисних засобів, планування щодо запобігання терористичних актів, стримування розвитку подій і, зрештою, зменшення їх наслідків. Атакуючи коли це не очікувано, використовуючи нестандартні засоби, терористи використовують слабкість шаблонно підготовлених збройних сил проти чітко визначеного ворога. Хоч уразливих місць і неможливо уникнути, ризики можна обмежити ідентифікацією цих слабких місць, оцінюванням ймовірності атаки, визначенням масштабів небажаних наслідків і шляхів їх ліквідації. Ключовим елементом ефективної антитерористичної стратегії є оцінювання підтвердження місць або можливих актів терористичного ризику. Вивлення загрози терористичних атак потребує збору інформації з різних джерел, більшість з яких містить невизначеності, які складно (або неможливо) розкрити. Саме тому для аналізу та прогнозування терористичних ризиків доцільно застосовувати мережі Байєса.

Дані щодо уразливих місць і шляхів ліквідації наслідків терористичних атак найчастіше ґрунтуються на досвіді або судженнях експертів з безпеки, а оцінювання наслідків потребує моделювання, виконуваного фахівцями. Антитерористичний планер кожної експертно-технічної групи відповідає за збір та обробку всієї інформації та врахування всіх можливих ризиків [202].

Site Profiler Installation Security Planner (ISP) – пакет такого програмного забезпечення на основі мереж Байєса. Дане програмне забезпечення комбінує дані з аналітичних моделей, симуляцій, історичних даних та суджень користувача і оцінює ризик терористичних атак, використовує об'єктно-орієнтовану базу даних для зберігання фрагментів мережі Байєса і керування інформацією щодо кожного окремого географічного місця.

Для оцінювання ризику користувач вибирає сценарії подій, які повинні бути оцінені, а система збирає для кожної можливої загрози відповідні фрагменти мереж Байєса в одну мережу. У Site Profiler така мережа називається Risk Influence Network (RIN) – мережа для аналізу впливів ризику (рис. 9.3).

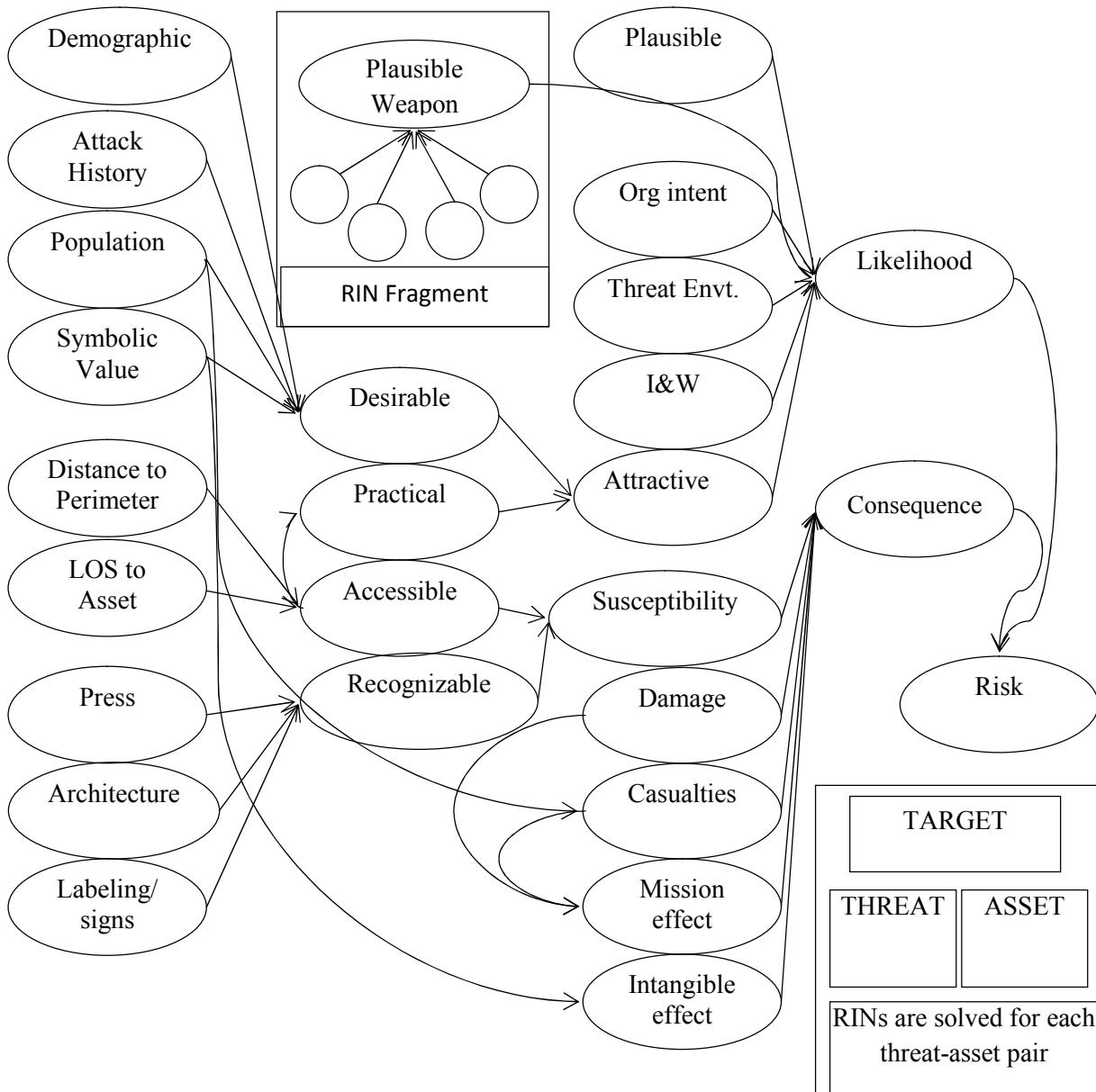


Рис. 9.3. Приклад RIN мережі в системі Site Profiler

Змінні, значення яких наявні у базі даних системи, стають спостереженнями для мережі, а змінні, які відповідають результатам моделювання – отримують шляхом моделювання без участі користувача. Додаткові судження та спостереження

вводяться користувачем через графічний інтерфейс програми. Система виконує розповсюдження значень всіх спостережень по мережі, обчислює значення шуканих змінних та повертає результат користувачеві.

Мережі Байєса застосовуються і в космічних дослідницьких програмах. Для центру керування польотів NASA Mission Control Center в Х'юстоні була розроблена система підтримки прийняття рішень Vista [203] із застосуванням мереж Байєса. Ця система аналізує телеметричні дані і в реальному часі ідентифікує, яку саме інформацію необхідно виводити на монітори [204]. Також мережа Байєса спільно з генетичними алгоритмами використовується для паралельних обчислень у системі OVERMIND, яка, в свою чергу представляє собою частину он-лайн експертної системи PESKI, призначеної для інженерно-технічної діагностики в космічній програмі Space Shuttle [137]. На рис. 9.4 зображена схема двигуна орбітальної системи маневрування, а на рис. 9.5 – мережа Байєса, що відповідає цій схемі.

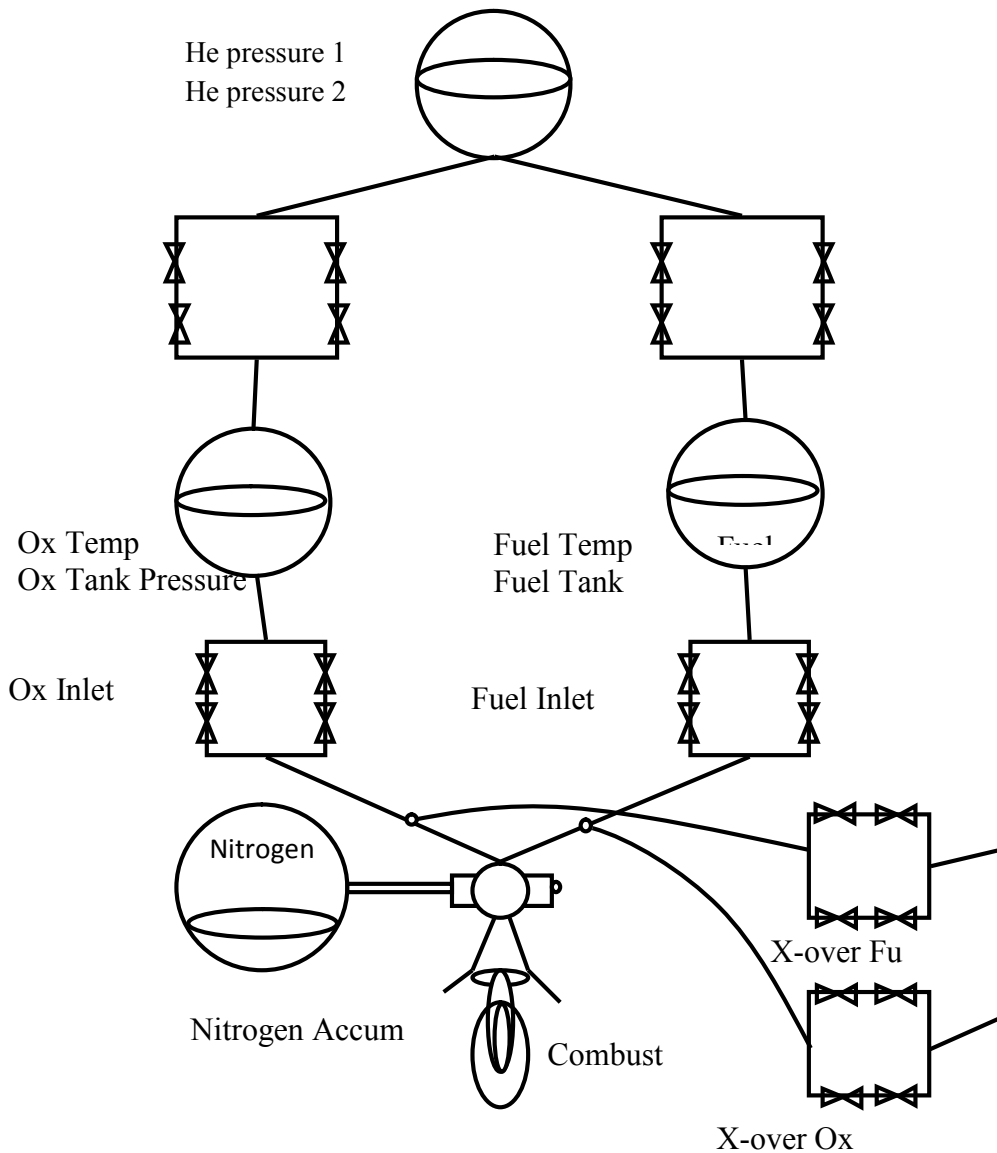


Рис. 9.4. Схема двигуна орбітальної системи маневрування

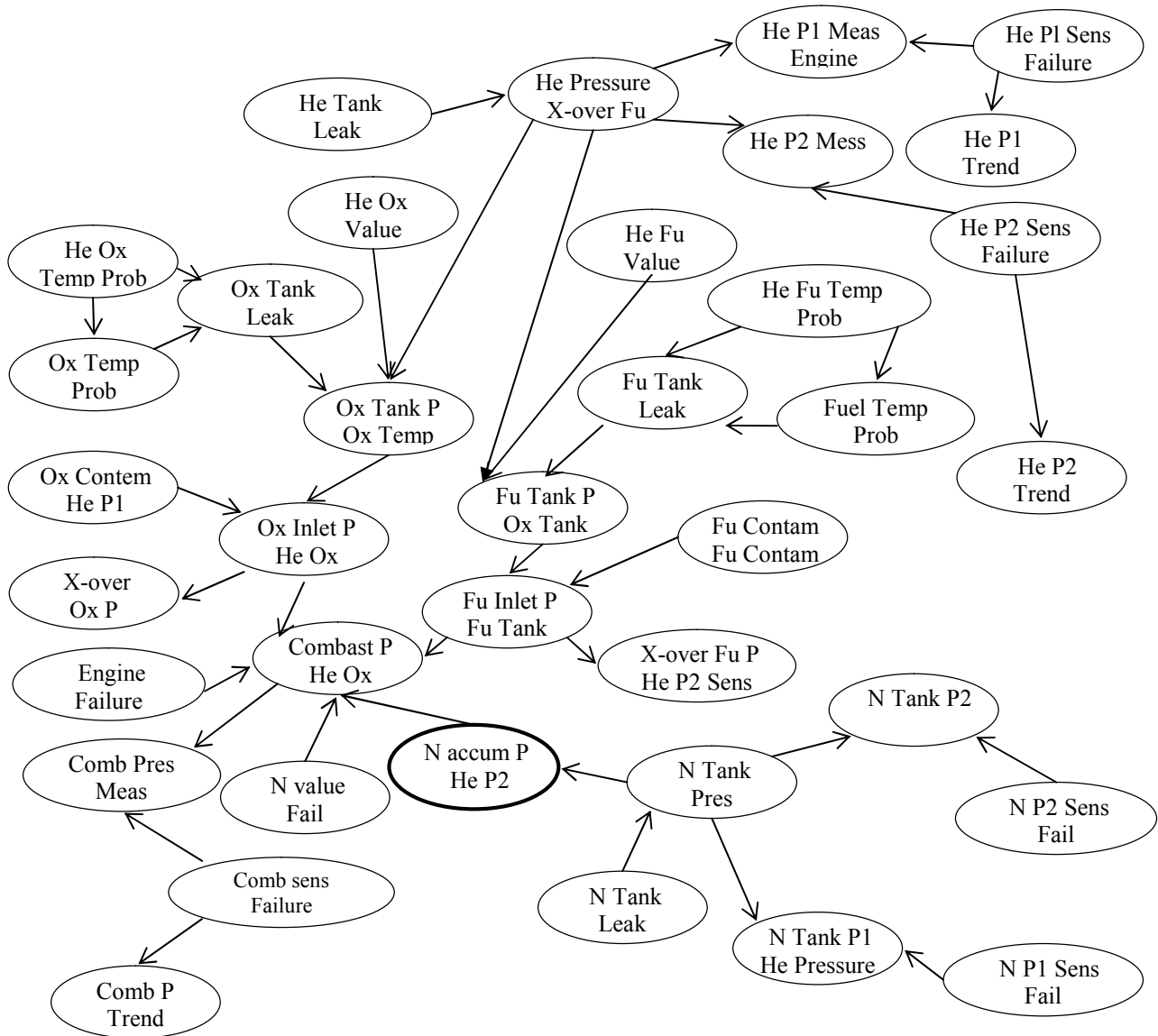


Рис. 9.5. Мережа Байєса двигуна орбітальної системи маневрування

Відділення Microsoft Research розробило Vista – систему на основі теорії прийняття рішень, що використовується в NASA Mission Control Center в Хьюстоні [206, 207]. В цій системі мережі Байєса використовуються для інтерпретації телеметричних даних в реальному часі. Vista забезпечує формування рекомендацій на основі правдоподібності щодо альтернативних поломок системи двигунів космічних шатлів та розробку плану дій з найвищою очікуваною корисністю. Більше того, система Vista застосовує методи теорії прийняття рішень для контролю функціонування діагностичних дисплеїв і в реальному часі ідентифікує, яку найбільш важливу інформацію необхідно виділити та представити.

Задачі діагностики надійності технічних систем також можуть бути вирішені із застосуванням мереж Байєса. Центр надійності програмного забезпечення при компанії Agena Ltd. ([www.agenaco.uk](http://www.agenaco.uk)) розробив TRACS (Transport Reliability Assessment and Calculation System) – систему прогнозування надійності військових транспортних засобів [205]. Даний програмний продукт застосовується Агенцією з розробок і оцінювання при міністерстві оборони Великобританії (DERA) для

оцінювання надійності транспортних засобів на всіх стадіях життєвого циклу їх проектування та розробки. Схема роботи системи TRACS представлена на рис. 9.6.

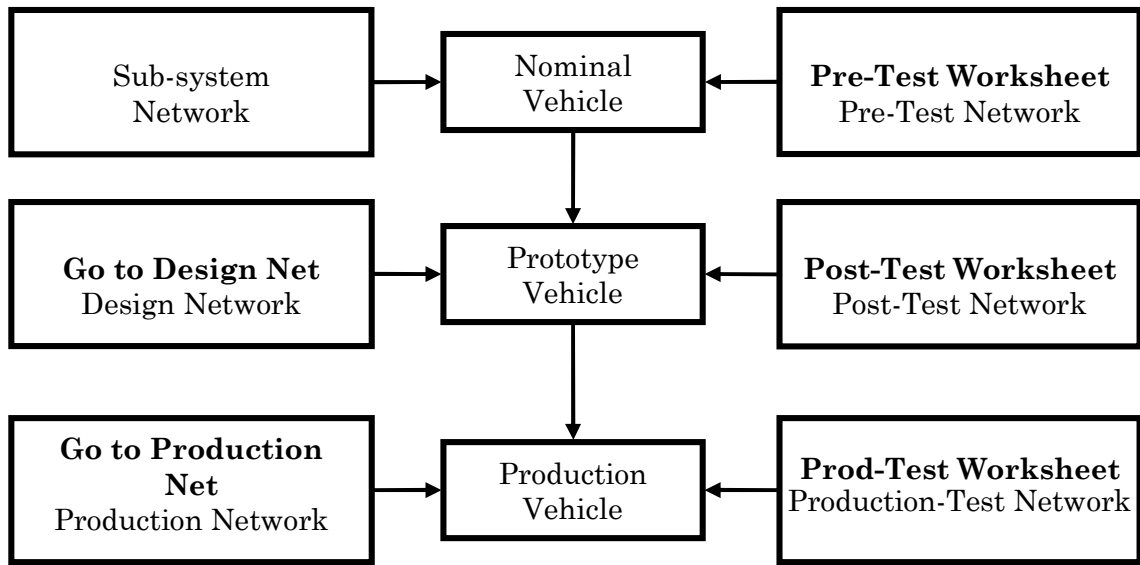


Рис. 9.6 Схема мережі Байєса системи TRACS та процес прогнозування

Сьогодні система TRACS експлуатується щоденно і застосовується DERA для розв'язання таких задач:

1. Ідентифікація найбільш перспективних транспортних засобів з числа тендерів перед початком розробки і тестування прототипів.
2. Обчислення надійності майбутнього високотехнологічного транспортного засобу на стадії формування вимог.
3. Зменшення об'єму ресурсів необхідних для тестування транспортних засобів на тестових треках.
4. Моделювання впливів низькоякісного проектування та виробничого процесу на надійність транспортного засобу.
5. Ідентифікація найбільш ймовірних наслідків ненадійності та виконання правил типу "що буде, якщо?" для аналізу та виявлення найкорисніших удосконалень процесу.

Система TRACS побудована на основі модульної архітектури і включає такі п'ять основних мереж Байєса (рис. 9.7):

1. Мережа оновлення, що використовується для прогнозування надійності підсистем на основі історичних даних про відмови аналогічних підсистем.
2. Рекурсивна мережа, що використовується для об'єднання ймовірнісних розподілів параметрів надійності підсистем з метою отримання рівня прогнозу для транспортного засобу.
3. Мережа якості проектування для оцінки ненадійності проекту, спричиненої низькоякісним процесом проектування.
4. Мережа якості виробництва для оцінки ненадійності, спричиненої низькоякісним процесом виробництва.
5. Мережа для тестування транспортних засобів, що використовує дані про відмови, отримані під час проведення випробувань, і робить висновок про надійність транспортного засобу.

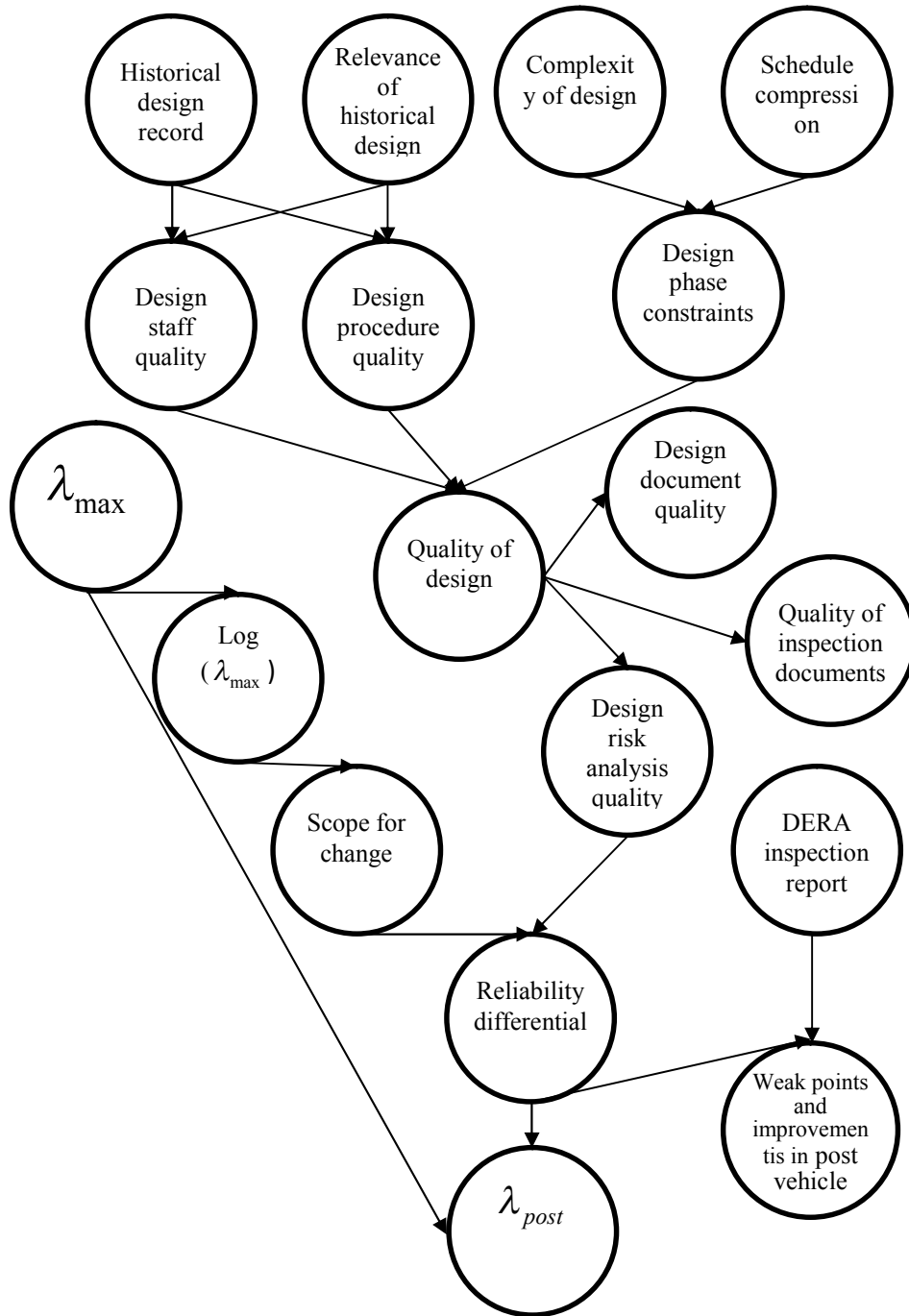


Рис. 9.7. Структура мережі Байєса для проектування, виробництва та тестування військових транспортних засобів

Докладнішу інформацію про систему можна знайти на сайті: [www.research.microsoft.com/research/dtg/horvitz/vista.htm](http://www.research.microsoft.com/research/dtg/horvitz/vista.htm).

Система TRACS може моделювати транспортні засоби з довільною достатньо великою кількістю підсистем. Кожна підсистема складається з більше, ніж мільйона комбінацій станів, згенерованих завдяки використанню ієрархічної байєсівської моделі із стандартними статистичними розподілами. Мережі якості проектування та



виробництва містять 35 вершин кожна, багато з яких мають умовні ймовірнісні розподіли, що прямо задаються інженерними експертами DERA.

Для керування транспортними засобами та діагностики відмов також ефективно використовуються мережі Байєса. Так, Automotive Information Systems (AIS) розробили більше 600 різних мереж Байєса для діагностики 15-и загальних автомеханічних проблем для близько 10000 різних транспортних засобів. Кожна мережа складається з не менше, ніж 100 вершин. Їх популярний програмний продукт AutoFix розроблено за допомогою пакета DX-press компанії Knowledge Industries, Inc. (KI). AutoFix – система для формування ймовірнісних висновків, покладена в основу додатку Diagnosis/SmartFix, який доступний на сайті [www.PartsAmerica.com](http://www.PartsAmerica.com). AIS та KI заявляють, що вони співпрацюють для розв'язання багатьох цікавих задач з метою відображення "промислової потужності" мереж Байєса.

Прикладом застосування мереж Байєса у галузі національної безпеки є інформаційно-аналітична система МВС України, розроблена О. М. Терентьевим на базі SAS технології в рамках науково-дослідницького проекту NUKR.SFPP G4877 "Моделювання та зменшення впливу соціальних лих, що виникли внаслідок катастроф та тероризму" (наукової програми НАТО "Наука заради миру та безпеки"). Система призначена для аналізу та прогнозування кількості злочинів за їх підтипами.

### ***Фінанси та економіка***

Фінансово-економічна діяльність будь-якого підприємства супроводжується накопиченням різноманітної бізнес-інформації, яка потребує обробки та прийняття рішень на основі цієї обробки. Так, компанія Hugin Expert A/S пропонує різні комерційні програмні комплекси для бізнесу. Наприклад, пакет програм Poulin-Hugin призначений для оцінювання ризиків та прогнозування прибутковості портфелів фінансових інструментів, а також інвестиційних проектів [113]. Основною перевагою мереж Байєса при вирішенні економічних задач є можливість одночасного врахування кількісних та якісних ринкових показників, динамічне врахування нової інформації, а також використання явної залежності між існуючими факторами, що впливають на зміну фінансових показників, а також наочність моделювання.

Компанія Data Digest [198] використовує мережі Байєса для моделювання та прогнозування поведінки споживачів у різноманітних бізнес-ситуаціях.

На застосуванні мереж Байєса в фінансовому секторі спеціалізується Bayesian Belief Network Application Group [199]. Однією з її відомих розробок є програмне забезпечення, що реалізує сегментацію клієнтів банку. Особливістю даної системи є те, що бізнес-правила сегментації, які застосовуються для класифікації клієнтів банку, попередньо визначаються з використанням підходу, що ґрунтується на правилах експертних систем. Bayesian Belief Network Application Group виконала реалізацію правил експертних систем з використанням мереж Байєса. Розроблена система отримала схвальні оцінки менеджменту банку Barclays.

Мережі Байєса застосовуються і для оцінки ризику кредиторів, для короткострокового прогнозування розвитку фінансово-економічних процесів.

**Комп'ютери і системне програмне забезпечення**

Сьогодні відомі застосування мереж Байєса для керування інтерфейсними агентами в системі Microsoft Office для діагностики проблем у функціонуванні принтерів, довідкових та wizard-системах. Наприклад, усім відома "скріпка-помічник" з редактора MS Word [25, 208]. Байєсівські мережі використовують також при створенні пошукових машин в Інтернеті для забезпечення можливості прогнозування модифікацій запитів, що надходять від користувачів. Також широко застосовуються мережі Байєса як антиспамовий фільтр у поштових програмах.

Пошукова система, в основу якої була покладена мережа Байєса використовується для обробки пошукових запитів користувачів правової інформації в парламенті Андалусії (Іспанія) [209]. Головною ціллю даної розробки було зробити доступними для населення обговорювані в парламенті законопроекти. На рис. 9.9 зображено приклад мережі Байєса, що представляє структурований документ.

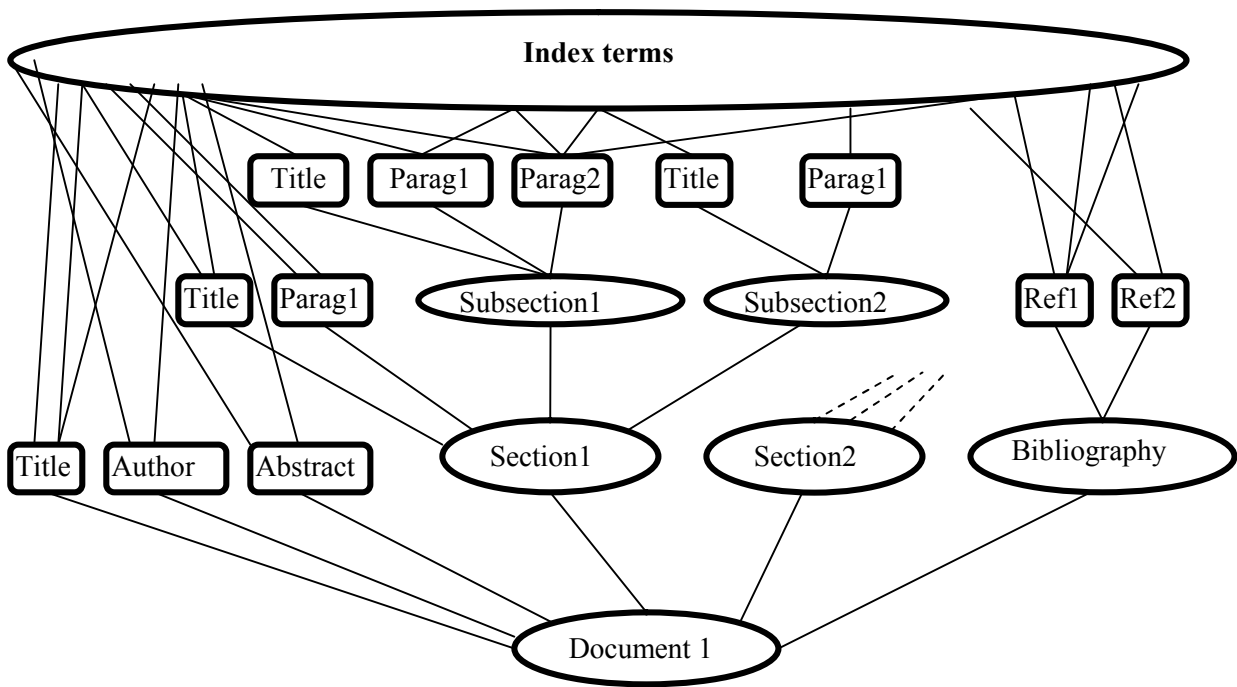


Рис. 9.9. Мережа Байєса структурованого документу

Ця система показала себе ефективною в роботі і є наочним прикладом того, як інформаційні технології, що ґрунтуються на мережах Байєса, можуть бути використані при розв'язанні задач, розмірності яких до цього часу не дозволяли їх вирішувати [202].

Іншим прикладом є система текстового пошуку SymText, розроблена Департаментом медичної інформатики при медичному університеті Юти. У даній системі мережі Байєса використовуються для розв'язання таких задач:

- представлення семантичних даних;
- зв'язування слів для відображення понять;
- вирішення неоднозначності компонентів значень і структури;
- відновлення термів, які пропущені через опущення елементів мови, помилки або знання, що отримуються з контексту;

– інших завдань обробки природної мови.

В ході реалізації системи SymText було створено чотири мережі, що містили від 14 до 30 вершин та від 3 до 4 шарів і які мали в середньому по 1000 ймовірнісних значень кожна. Кожна мережа моделює "контекст" інформації, позначеної для пошуку. Три мережі мають просту структуру дерева, а одна використовується для моделювання різниці між позитивними і негативними мовними структурами. Розробники запевняють, що система дуже корисна, проте має дві проблеми:

- створення мережі є дорогим та вимагає багато часу;
- для формування ймовірнісного висновку в мережі необхідні значні обчислювальні ресурси.

Мережі Байєса застосовуються також і для відновлення 3D сцен за динамічною 2D інформацією та синтезу статистичних зображень високої чіткості з відеосигналу [210].

Знаходять своє застосування мережі Байєса і в комп'ютерних іграх. Так, Валадарес [201] розробив комп'ютерну гру, що моделює розвиток симульованого світу з використанням мережі Байєса.

При побудові системи обробки дефектів процесорів компанія Intel ([www.intel.com](http://www.intel.com)) також використала мережу Байєса. Принцип обробки полягав у тому, що саме остаточні тести на напівпровідникових чипах видають можливі проблеми процесорів. Інженери та науковці Intel розпочали розробку системи діагностики ще в 1990-х, і після багатьох років "еволюції", система працює досить стабільно. Мережа має три рівні і декілька сотень вершин. Однією з проблем даної мережі було оцінювання та налаштування значень апріорних ймовірностей, тому новіші частини діагностичної системи зараз розробляються з використанням нечіткої логіки, яка виявилась зручнішою для побудови та налаштування.

Іншим напрямком використання мереж Байєса є системи розпізнавання зображень та звуку. Зокрема, група дослідників "машинного зору" університетів Редінга та Лідса (Великобританія) розробила інтегровану систему технічного зору для транспорту та пішоходів ([www.cvg.cs.rdg.ac.uk/~imv](http://www.cvg.cs.rdg.ac.uk/~imv)). А Фам [211] розробив систему розпізнавання обличчя з використанням мереж Байєса.

Також динамічні байєсівські мережі знайшли застосування для розпізнавання голосу або мовних сигналів [212], розроблена система для аудіо-візуального розпізнавання голосу [213]. Більш детальна інформація про дані розробки представлена на сайтах компаній [214, 215].

## **Освіта**

Навчальний дослідницький центр при університеті Пітсбурга (США) розробив інтелектуальну навчальну систему Andes ([www.pitt.edu/~vanlehn/andes.html](http://www.pitt.edu/~vanlehn/andes.html)) для вивчення фізики. Andes формує план того, як студенти мають опанувати курс фізики та з часом відслідковує і оцінює рівень засвоєння знань студентами.

Роялті та інші розробили систему РОЕТ – навчальний консультаційний інструмент, що моделює структуру навчальних курсів для студентів та відповідних академічних довідок [135] ([www.cs.uky.edu](http://www.cs.uky.edu)). Більшість змінних представляють собою оцінки по курсах і безпосередньо набувають значення оцінок, а також значення "Відсутній" та "Анульовано".

## 9.2. Огляд комп'ютерних програм для створення та застосування мереж Байєса

Стрімкий розвиток досліджень мереж Байєса протягом останніх 20 років супроводжувався зростанням кількості програмних засобів, призначених для роботи з ними. Ці інструменти розроблювались як для розв'язання дослідницьких задач, так і для практичного застосування мереж Байєса в різноманітних областях діяльності людини.

Перші програмні продукти для створення мереж Байєса у формі більшій, ніж просто реалізація алгоритмів, з'явилися на ринку наприкінці 80-х років XX століття. Одним із перших програмних продуктів даного типу є програма Hugin, розроблена в університеті Алборга, Данія. Спочатку програмне забезпечення Hugin було безкоштовним, але згодом воно перетворилось на широко використовуваний комерційний продукт. Програмним продуктом на основі мови Lisp була розроблена Роквеллом експериментальна модель IDEAL (Influence Diagram Evaluation and Analysis in Lisp). Ще одним із перших інструментів для формування висновку в байєсівських мережах, що заслуговує на увагу, є CABeN (a Collection of Algorithms for Belief Networks). Цей програмний продукт містив бібліотеку функцій для різних алгоритмів ймовірнісного висновку на основі стохастичного моделювання.

Пізніше були розроблені такі програмні продукти, як Analytica (виробник – компанія Lumina Decision Systems, Inc., заснована Максом Хенріоном та Брайаном Арнольдом в 1991 році) та Netica (зараз виробляється компанією Norsys Ltd).

Сьогодні розроблено та активно використовується значна кількість різноманітних програмних продуктів, що реалізують мережі Байєса. Більш детальний їх перелік представлений у додатках посібника, а також у мережі Інтернет, зокрема за посиланням: <http://directory.google.com/Top/Computers/ArtificialIntelligence>, а також у BayesianNetworkRepository – сховищі, що містить приклади мереж Байєса та множини даних для їх навчання (<http://www.cs.huji.ac.il/labs/combio/Repository>).

Однак, на описі деяких програмних продуктів, призначених для роботи з мережами Байєса слід зупинитись детальніше.

**BayesiaLab** (<http://www.bayesia.com/>) дуже зручна у використанні програма, розроблена французькою компанією Bayesia SA. У програму інтегровано декілька методів для автоматичної побудови структури мережі Байєса та формування ймовірнісного висновку, особливий інтерес викликає швидкий та зручний метод SopLeq. Ціна програми залежить від типу ліцензії та комплектації і коливається від 190 € за найдешевшу академічну версію до 88000€ за повну ліцензію на професійну версію. Існує можливість користування програмою безкоштовно протягом місяця, але в цьому випадку немає можливості зберігати результати роботи.

**Netica** (<http://www.norsys.com/>) – програмний продукт, розроблений компанією Norsys Software Corporation, Ванкувер (Канада), що спеціалізується на розробці програмного забезпечення для байєсівських мереж. Програма Netica – основне досягнення компанії розробляється з 1992 року, комерційно доступна з 1995 року. Нині Netica є одним з найбільш використовуваних експертами інструментів для створення мереж Байєса та отримання ймовірнісного висновку. Недоліком програми є відсутність можливості автоматичної побудови мережі за навчальними даними. Вартість академічної версії програми складає \$285, а комерційної – \$685. Також

доступна безкоштовна версія програмного забезпечення, але в цьому випадку максимально можливий розмір мережі Байєса для роботи та зберігання результатів обмежується 15 вершинами, а максимально можливий розмір вибірки даних складає 999 записів.

**Hugin-Expert** (<http://www.hugin.com/>) розроблена компанією Hugin Expert A/S, Ольсборг (Данія). Її основний продукт Hugin почав створюватися під час робіт за проектом ESPRIT для проблеми діагностики нервово-м'язових захворювань. Потім почалася комерціалізація результатів проекту і основного інструменту – програми Hugin. Сьогодні програмне забезпечення Hugin адаптоване в багатьох дослідницьких центрах компанії в 25 різних країнах і використовується в сферах, пов'язаних з аналізом та підтримкою прийняття рішень, прогнозуванням, діагностикою, управлінням ризиками та оцінюванням безпеки технологій.

Hugin є програмною реалізацією системи підтримки прийняття рішень на основі мереж Байєса, має дві версії: Pro і Explorer, має працювати у двох основних режимах: редагування і побудови мережі Байєса, а також заповнення таблиць умовних ймовірностей і побудови ймовірнісного висновку. Ціна програмного забезпечення, залежно від типу ліцензії, коливається від 6426 до 50999 данських крон, є можливість використання безкоштовної версії програми, у якій немає можливості автоматичної побудови мережі Байєса.

**Bayes Net Toolbox for Matlab** (<http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html>), розроблена професором Кевіном Мерфі для системи Matlab, є безкоштовною. Всі модулі програми доступні користувачу. Існує можливість доповнювати програму своїми методами та модулями, але необхідно мати навички роботи в пакеті Matlab та розуміти теорію побудови мереж Байєса. Серед недоліків слід зазначити відсутність зручного графічного інтерфейсу та можливості візуалізації результатів.

**MSBNx** (<http://research.microsoft.com/adapt/MSBNx/>) – безкоштовна програма, розроблена компанією Microsoft. Основним її недоліком є відсутність можливості автоматичної побудови мережі Байєса за навчальними даними.

**GeNIe** (<http://genie.sis.pitt.edu/>), яке є ще одним із безкоштовних програмних продуктів, розроблена лабораторією систем підтримки прийняття рішень Пітсбургського Університету США. Програмне забезпечення GeNIe дуже зручне у використанні, має можливість автоматичної побудови мереж Байєса та підключення власних модулів з алгоритмами, написаними на мові програмування C.

**Bayesware Discoverer** (<http://bayesware.com/>) - програмне забезпечення розроблене англійською компанією Bayesware. Серед його особливостей слід відзначити можливість автоматичної побудови мереж Байєса за навчальними даними, симуляційне порівняння декількох моделей, побудованих на основі мереж Байєса, використання алгоритму Bound and Collapse для випадку неповних даних, заповнення таблиці умовних ймовірностей та побудови ймовірнісного висновку, роботи з категоріальними даними та неперервними змінними. Однак дане програмне забезпечення досить дороге. Вартість ліцензії складає від \$3000 до \$8500, хоча протягом 30 днів програму можна використовувати безкоштовно без будь-яких обмежень функціональності.

**Naive Bayes** (<http://www.oracle.com>) є невід'ємною складовою Oracle Data Mining, що дозволяє обробляти великі масиви даних та в автоматичному режимі будувати моделі та прогнози на основі мереж Байєса, підєднувати власні модулі та

використовувати дані будь-яких типів та форматів. Однак, дане програмне забезпечення досить дороге – вартість ліцензії складає від \$9000.

**SAS Enterprise Miner** – спеціалізований інструмент призначений для спрощення процесу аналізу даних, при створенні високоточних інтелектуальних і описових моделей, заснованих на великих обсягах даних.

Сучасна версія програми SAS Enterprise Miner 13.2 на основі ядра системи SAS Foundation 9.4 відповідає сучасним вимогам щодо обробки великих обсягів даних завдяки реалізації алгоритмів машинного навчання, наприклад, нейронні мережі, кластерний аналіз, дерева рішень, ансамблю моделей, градієнтний бустінг, мережі Байєса, регресійний аналіз та інші. Це рішення є масштабованим у відповідності до розміру розв'язуваних бізнес-проблем. Його модуль високопродуктивної обробки даних використовує технології багатопотокової та розподіленої обробки даних, використовуючи наявні процесорні можливості та пам'ять комп'ютера. Також є можливість використовувати дану технологію на Hadoop кластерах, таких як Cloudera або Hortonworks, або на виділеному апаратному забезпеченні від Teradata, Pivotal та Oracle. Для підвищення продуктивності в SAS 9.4 реалізовано можливість програмування, зокрема мови програмування R [3].

### 9.3. Застосування сучасних програмних продуктів для побудови мереж Байєса на прикладі SAS Enterprise Miner

Для побудови мереж Байєса в SAS Enterprise Miner 13.2 застосовується процедура **HPBNET** (потребує наявності ліцензії **SAS High-Performance Data Mining** [219]) - високопродуктивною процедурою, що дозволяє будувати мережу Байєса на основі вхідних даних. Процедура **HPBNET** реалізує різноманітні типи мереж Байєса: наївна мережа Байєса, дереводоповнена мережа Байєса (Tree-Augmented Network – **TAN**), наївна збільшена мережа Байєса (**BAN**), структура батько-дитина (Parent-Child – **PC**) та ковдра Маркова (**Markov Blanket**). Процедура **HPBNET** робить вибір змінних на основі проведення незалежних тестів, а також автоматично обирає найкращу модель. Окрім цього створює код кроку даних, що в подальшому використовується для їх скорінгування.

Процедура **HPBNET** є високопродуктивною аналітичною процедурою, яка дозволяє:

- виконувати розподілені обчислення на обчислювальних кластерах;
- працювати за технологією клієнт-сервер, коли основні обчислення робить сервер;
- використовувати всі ядра та потоки комп'ютера, відповідно до заданого режиму роботи;
- виконувати структурне навчання на основі використання ефективних локальних алгоритмів;
- обирати вхідні змінні на основі використання тестів на умовну незалежність;
- автоматично обирати найкращі параметри із використанням валідаційних вибірок;
- будувати різноманітні типи мереж Байєса;
- використовувати як інтервальні, так і номінальні змінні;
- перетворювати інтервальні змінні в номінальні;
- працювати з пропусками в даних;

- мультипоточкова обробка даних.

Всі коди програм та набори даних для роботи, що використовуються в даному розділі представлені на сайті <https://sites.google.com/site/data4mining/datasets/book3>.

### ***Побудова програмного коду для формування вхідного набору даних***

В якості прикладу розглянемо задачу використання лікарського засобу анальгетика (речовина призначена для зняття болю) в неврології. Виконується порівняння двох видів ліків та плацебо. В якості змінної-відгуку розглядається звіт про наявність або відсутність болю. Вибірка (Neuralgia) складається з даних про 60 пацієнтів.

Для формування вхідного набору даних Neuralgia, використовуються наступний **код програми на SAS Base**:

```
Data Neuralgia;
input Treatment $ Sex $ Age Duration Pain $ @@;
datalines;
P F 68 1 No B M 74 16 No P F 67 30 No
P M 66 26 Yes B F 67 28 No B F 77 16 No
A F 71 12 No B F 72 50 No B F 76 9 Yes
A M 71 17 Yes A F 63 27 No A F 69 18 Yes
B F 66 12 No A M 62 42 No P F 64 1 Yes
A F 64 17 No P M 74 4 No A F 72 25 No
P M 70 1 Yes B M 66 19 No B M 59 29 No
A F 64 30 No A M 70 28 No A M 69 1 No
B F 78 1 No P M 83 1 Yes B F 69 42 No
B M 75 30 Yes P M 77 29 Yes P F 79 20 Yes
A M 70 12 No A F 69 12 No B F 65 14 No
B M 70 1 No B M 67 23 No A M 76 25 Yes
P M 78 12 Yes B M 77 1 Yes B F 69 24 No
P M 66 4 Yes P F 65 29 No P M 60 26 Yes
A M 78 15 Yes B M 75 21 Yes A F 67 11 No
P F 72 27 No P F 70 13 Yes A M 75 6 Yes
B F 65 7 No P F 68 27 Yes P M 68 11 Yes
P M 67 17 Yes B M 70 22 No A M 65 15 No
P F 67 1 Yes A M 67 10 No P F 72 11 Yes
A F 74 1 No B M 80 21 Yes A F 69 3 No
;
```

run;

Вхідні змінні набору даних Neuralgia.

**Pain** – цільова змінна, що приймає значення:

- Yes – якщо пацієнт відчуває біль;
- No – в протилежному випадку.

**Treatment** – вхідна змінна номінального типу, що приймає значення:

- A – лікарський засіб 1;
- B – лікарський засіб 2;
- C – плацебо.

**Sex** – змінна, що описує стать пацієнта, приймає значення:

- M – чоловік;

- F – жінка.

**Age** – змінна, що описує вік пацієнта, в якому почалося лікування.

**Duration** – змінна, що відповідає терміну в місяцях, протягом якого пацієнт відчував біль, до початку лікування.

### ***Навчання структури мережі***

Наступний код програми призначений для побудови мережі Байєса із використанням номінальних змінних **Treatment** та **Sex**, а також інтервальних змінних **Age** та **Duration**. Для побудови структури мережі інтервальні змінні перетворюються в номінальні за принципом розбиття на три діапазони з інтервалом рівної довжини.

#### ***Код програми:***

```
proc hpbnet data=Neuralgia numbin=3;  
TARGET Pain;  
input Treatment Sex/level=NOM;  
input Age Duration/level=INT;  
output network=network varselect=vartest;  
run;
```

Для відображення отриманої структури мережі Байєса використовується наступний код програми:

```
proc print data=network noobs label;  
var _parentnode_ _childnode_;  
where _type_="STRUCTURE";  
run;
```

#### ***Результати роботи програми:***

```
Parent Child  
Node Node
```

```
Treatment Pain  
Pain Age
```

### ***Виведення таблиць умовних ймовірностей***

Наступний фрагмент коду програми дозволяє виводити таблиці умовних ймовірностей для кожної вершини мережі. Ці таблиці умовних ймовірностей можна використовувати як для скорінгування, так і для побудови ймовірнісного висновку в мережі.

#### ***Код програми:***

```
proc print data=network noobs label;  
var _parentnode_ _parentcond_ _childnode_ _childcond_ _value_;  
where _type_="PROBABILITY";  
run;
```



**Результат виконання програми.**

Parent Node	Parent Condition	Child Node	Child Condition	Value
Treatment A				0.33333
Treatment B				0.33333
Treatment P				0.33333
Treatment A	Pain	YES		0.27273
Treatment A	Pain	NO		0.72727
Treatment B	Pain	YES		0.27273
Treatment B	Pain	NO		0.72727
Treatment P	Pain	YES		0.72727
Treatment P	Pain	NO		0.27273
Pain	YES	Age	<67	0.17857
Pain	YES	Age	<75	0.35714
Pain	YES	Age	>=75	0.46429
Pain	NO	Age	<67	0.31579
Pain	NO	Age	<75	0.60526
Pain	NO	Age	>=75	0.07895

**Вибір змінних**

У результаті роботи процедури *HPBNET* була побудована наступна структура мережі:

```

Parent Child
Node Node

Treatment Pain
Pain Age
    
```

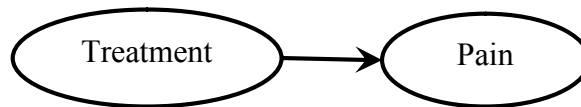


Рис. 9.10. Структура мережі Байєса "батько-дитина", побудована на основі набору даних Neurologia

Як видно з рис. 9.10, до структури не були включені змінні **Sex** та **Duration**, тому що процедура *HPBNET* робить вибір змінних автоматично на основі використання тестів на незалежність. Код програми, що показує результати вибору змінних є наступним.

**Код програми:**

```

proc print data=vartest noobs label;
run;
    
```

Результати роботи програми представлені у табл. 9.2.

## Результати вибору змінних

Variable Name	Selected	Chi-Square Statistics	G-Square Statistics	P-Value of Chi-Square Statistics	P-Value of G-Square Statistics	Mutual Information	Degree of Freedom	Conditional Variables
Sex	NO	7.2000	7.5945	0.06579	0.05518	0.34481	3	Treatment
Treatment	YES	13.7143	14.0230	0.00105	0.00090	0.45652	2	
Age	YES	14.6000	15.2712	0.00068	0.00048	0.47404	2	
Duration	NO	2.2579	3.3485	0.32337	0.18745	0.23298	2	

Змінна **Duration** була виключена зі структури мережі, тому що  $p$ -значення критеріях  $\chi$ -квадрат та  $G$ -квадрат статистика більше за 0,05 (критичне значення **ALPHA** за замовчуванням). Змінна **Sex** також має умовну незалежність від **Pain** через змінну **Treatment** (тому що при більш прискіпливому аналізі **Treatment** впливає на **Sex**), як наслідок процедура **HPBNET** виключає її зі структури мережі.

**Синтаксис процедури HPBNET**

Синтаксис процедури **HPBNET** має наступний вигляд.

```
PROC HPBNET < options > ;
  TARGET variable </ORDER=order> ;
  INPUT variables </ LEVEL=level > ;
  ID variables ;
  FREQ variable ;
  PARTITION < options > ;
  PERFORMANCE < options > ;
  OUTPUT < options > ;
  CODE FILE='external-file' | fileref ;
```

**TARGET** – цільова змінна.

**INPUT** – вхідна змінна або декілька змінних.

Опції PROC HPBNET.

```
PROC HPBNET < options > ;
```

## Опції даних процедури HPBNET

Опція	Опис
DATA	Вхідний набір даних.
NUMBIN	Кількість інтервалів, на які розбивається інтервальна змінна.
PRESCREENING	Попередні вхідні змінні.
VARSELECT	Вибір вхідних змінних
MISSINGINT	Налаштування режиму обробки відсутніх значень в інтервальних змінних.
MISSINGNOM	Налаштування режиму обробки відсутніх значень в номінальних змінних.

Таблиця 9.4

Опції побудови структури мережі процедури **HPBNET**

Опція	Опис
STRUCTURE	Тип структури мережі.
PARENTING	Метод побудови структури мережі.
MAXPARENTS	Максимальна кількість батьків, що можуть бути у кожній вершині.

Таблиця 9.5

Опції вибору моделі та побудови тестів на незалежність процедури **HPBNET**

Опція	Опис
BESTMODEL	Вибір найкращої моделі.
INDEPTEST	Налаштування методів тестування на незалежність.
ALPHA	Критичне значення тесту на незалежність із використанням статистики $\chi$ -квадрат або $G$ -квадрат.
MIALPHA	Критичне значення тесту на незалежність із використанням значення взаємної інформації.

Розглянемо більш детально застосування окремих опцій процедури **HPBNET**

▪ **ALPHA= число**

Завдання значимого рівня тесту на незалежність із використанням статистик  $\chi$ -квадрат або  $G$ -квадрат. Область допустимих значень від 0 до 1, включно. Якщо треба обрати модель з декількох, можна задати до п'яти значень, розділені пропусками (spaces). Якщо задано декілька значень, але не обрана опція **BESTMODEL**, то процедура **HPBNET** використає перше значення, а інші проігнорує. По замовчанню **ALPHA = 0.05**.

▪ **BESTMODEL**

Найкраща модель буде обрана із використанням валідаційного набору даних, який можна задати за допомогою оператора **PARTITION**. Якщо задана опція **BESTMODEL**, то можна налаштувати множинні значення **ALPHA=**, **PRESCREENING=**, **VARSELECT=**, **STRUCTURE=** та **PARENTING=**. Процедура **HPBNET** використовує похибку помилкової класифікації на валідаційних даних автоматично. За замовчуванням опція **BESTMODEL** не активована.

▪ **DATA= вхідний набір даних**

Задається ім'я набору даних, що містить навчальні дані, а також, можливо, валідаційні. За замовчуванням використовується найбільш новий створений набір даних.

▪ **INDEPTEST=ALL | CHIGSQURE | CHISQUARE | GSQUARE | MI**

Налаштування методу тесту на незалежність. Можуть бути використані наступні режими.

1) **INDEPTEST=ALL**

Всі статистики виконуються тести  $\chi$  - квадрат,  $G$  – квадрат, а також нормалізований тест на основі взаємної інформації. Змінна вважається незалежною від цільової, якщо  $p$ -значення статистик  $\chi$ -квадрат та  $G$ -квадрат більші за задане ALPHA та значення нормалізованого тесту взаємної інформації менше за задане значення MIALPHA.

**2) INDEPTEST=CHIGSQURE**

Використовуються обидві статистики  $\chi$ -квадрат та  $G$ -квадрат тестів на незалежність. Значення незалежне від цільової змінної, якщо обидва  $p$ -значення статистик  $\chi$ -квадрат, та  $G$ -квадрат більші за ALPHA.

**3) INDEPTEST= CHISQUARE**

Використовується статистика  $\chi$ -квадрат для тесту на незалежність. Змінна незалежна від цільової, якщо  $p$ -значення статистики більше за ALPHA.

**4) INDEPTEST= GSQUARE**

Використовується статистика  $G$ -квадрат для тесту на незалежність. Змінна незалежна від цільової, якщо  $p$ -значення статистики більше за ALPHA.

**5) INDEPTEST= MI**

Використовується нормалізований тест на незалежність на основі взаємної інформації. Змінна незалежна від цільової, якщо нормалізоване значення взаємної інформації менше за MIALPHA.

За замовчуванням використовується **INDEPTEST=CHIGSQURE**.

▪ **MAXPARENTS= ціле число**

Максимальна кількість батьківських вершин для кожної вершини-нащадка в структурі мережі. Може набувати значення від 1 до 16 включно. Якщо задана опція BESTMODEL, то процедура **HPBNET** починає підбір від 1 та приймає рішення щодо найкращого числа батьківських вершин. За замовчуванням MAXPARENTS=5.

▪ **MIALPHA= число**

Задає значення порогу тесту на незалежність на основі значення взаємної інформації. Діапазон допустимих значень від 0 до 1 включно. За замовчуванням MIALPHA = 0.05.

▪ **MISSINGINT=IGNORE | IMPUTE**

Налаштування для обробки відсутніх інтервальних змінних, є можливість задати наступні значення:

**IGNORE** – ігнорувати спостереження, що містять пропущені значення для будь-яких інтервальних змінних.

**IMPUTE** – замінювати пропуски для будь-яких інтервальних змінних математичним сподіванням.

За замовчуванням використовується режим **MISSINGINT=IGNORE**.

**MISSINGNOM=IGNORE | IMPUTE | LEVEL**

Налаштування для обробки пропусків номінальних змінних, є можливість задати наступні значення:

1) **IGNORE** – ігнорувати спостереження, що містять пропущені значення для будь-яких номінальних змінних.

2) **IMPUTE** – замінювати пропуски для будь-яких номінальних змінних модальним значенням.

3) **LEVEL** – використовувати відсутнє значення як окреме значення (стан) номінальної змінної.

За замовчуванням використовується режим **MISSINGINT=IGNORE**.

**NUMBIN**= ціле число.

**NBIN**= ціле число.

При розподілі інтервальної змінної, процедура **HPBNET** розподіляє її на рівні інтервали, кількість інтервалів може задаватися від 2 до 1024, включно. За замовчуванням **NUMBIN=5**.

### Оператор **STRUCTURE**

**STRUCTURE=MB | NAIVE | PC | TAN**

Сумісно з опцією **MAXPARENTS**, цей параметр задає структуру мережі, що буде побудована на навчальних даних. Можуть бути задані наступні значення:

1) **STRUCTURE=MB** – ковдра Маркова (*Markov Blanket*) для цільової вершини. Ковдра Маркова включає батьківські вершини, нащадки, а також батьків вершин-нащадків. Після навчання ковдри Маркова, процедура **HPBNET** визначає батьків вершини-нащадка, а також зв'язки між вершинами-нащадками. При цьому для режиму **STRUCTURE=MB** враховуються значення опцій **PRESCREENING** та **VARSELECT**.

2) **STRUCTURE=NAIVE** – структура у вигляді наївної мережі Байєса. У цьому випадку цільова вершина має спрямований зв'язок у напрямку кожної вхідної змінної. Якщо **MAXPARENTS=1**, то будується структура наївної мережі Байєса, якщо **MAXPARENTS** більше за 1, то будується структура розширеної наївної мережі Байєса (BAN).

3) **STRUCTURE=PC** – структура мережі батько-дитина (Parent-Child). PC-структура відрізняється від наївної тим, що, декілька вхідних вершин можуть бути використані в якості батьківських по відношенню до цільової змінної. Окрім цього, зв'язки від батьків до нащадків можуть також проходити через інші вершини-нащадки.

4) **STRUCTURE=TAN** – структура у вигляді доповненого дерева наївної мережі Байєса. Ця структура включає прямі зв'язки від цільової змінної у напрямку вхідної змінної за деревовидною структурою. Для того, щоб обрати найкращу структуру, можна задати декілька значень, у будь-якому порядку, розділивши їх пропусками та задати опцію **BESTMODEL**.

Якщо було задано декілька структур, але не задана опція **BESTMODEL**, процедура **HPBNET** використовуватиме перше значення і проігнорує інші. За замовчуванням використовується **STRUCTURE=PC**.

### Оператор **PARENTING**.

**PARENTING=BESTONE | BESTSET**

Для налаштування алгоритму для орієнтування мережі, структурі можна задати наступні значення:

1) **PARENTING=BESTONE** – використовується "жадібний підхід" для визначення батьків кожної вершини. На кожній ітерації в якості батьківської вершини додається найкраща вершина-кандидат.

2) **PARENTING= BESTSET** – на відміну від попереднього підходу, визначається одразу множина батьківських вершин серед можливих кандидатів-множин. Одразу можна використовувати обидва підходи, вказавши їх послідовно, розділивши пропусками (*space*). При цьому повинна бути вказана опція **BESTMODEL**. У випадку, коли опція **BESTMODEL** не вказана, буде використаний підхід, що вказаний першим. За замовчуванням **PARENTING=BESTSET**.

### Оператор **PRESCREENING**

**PRESCREENING=0 | 1**

Принцип первісного відбору змінних передбачає, що можуть бути задані наступні значення:

1) **PRESCREENING = 0** – використовуються всі змінні.

2) **PRESCREENING = 1** – використовуються тільки ті змінні, що залежать від цільової.

Також є можливість одночасно задати обидва режими. Серед них буде обраний найкращий: **PRESCREENING= 1 0**, але при цьому повинна бути задана опція **BESTMODEL**. Якщо задане значення **PRESCREENING= 1 0**, але без опції **BESTMODEL**, то буде обране значення, що записане першим. За замовчуванням **PRESCREENING=1**.

### Оператор **VARSELECT**

**VARSELECT=0 | 1 | 2 | 3**

Даний оператор визначає вибір вхідних змінних за межами їх первісного відбору. Може приймати наступні значення:

1) **VARSELECT=0** – використовує всі вхідні змінні, що залишаються після попереднього відбору, заданого опцією **PRESCREENING**.

2) **VARSELECT=1** – виконує тести на умовну незалежність кожної вхідної змінної по відношенню до цільової змінної.

3) **VARSELECT=2** – проводить тести на умовну незалежність кожної вхідної змінної по відношенню до цільової змінної по заданій множині змінних.

4) **VARSELECT=3** – визначає ковдру Маркова для цільової змінної.

Якщо задано **VARSELECT=1, 2** або **3**, то процедура **HPBNET** автоматично тестує кожну вхідну змінну на безумовну залежність цільової змінної відповідно до налаштувань опції **PRESCREENING**. Якщо не залишилося змінних, що задовольняють заданому рівню, то процедура **HPBNET** відкатується до попереднього рівня. Наприклад, якщо **VARSELECT=3** і не має змінних для коври Маркова, то процедура **HPBNET** повертається до **VARSELECT=2**.

Якщо потрібно обрати найкращу модель серед різних рівнів вибору змінних, то можна задати комбінацію значень для опції **VARSELECT** та опцію **BESTMODEL**. Якщо задано декілька значень для **VARSELECT**, але не задана опція **BESTMODEL**, то процедура **HPBNET** використовує лише перше значення та ігнорує наступні.

### Оператор **CODE**

**Синтаксис оператора:**

**CODE FILE='external-file' | fileref ;**

Оператор **CODE** є необов'язковим. Він дозволяє зберегти скорінговий код побудованої моделі у зовнішньому файлі, який у подальшому може бути використаний для скорінгування нових даних. Цей оператор можна використовувати декілька разів.

### Оператор **FREQ**.

#### **Синтаксис оператора:**

**FREQ** variable ;

Змінна, що вказується в операторі **FREQ**, ідентифікує чисельну змінну в наборі даних, що містить частоту появи кожного спостереження. Якщо для відповідного спостереження значення частоти не є цілим числом, то воно заокруглюється до цілого. Якщо значення частоти менше за 1 або є пропуском, то відповідне спостереження ігнорується. Коли для оператора **FREQ** не назначена змінна, то кожне спостереження має частоту появи, що дорівнює 1.

### Оператор **ID**.

#### **Синтаксис оператора:**

**ID** variables ;

Оператор **ID** може містити одну або декілька змінних як інтервального, так і номінального типів, що використовуються як ідентифікатор.

### Оператор **INPUT**

#### **Синтаксис оператора:**

**INPUT** variables </ LEVEL=level > ;

Оператор **INPUT** задає одну або декілька змінних в якості вхідних. Цей оператор можна використовувати декілька разів.

**LEVEL=NOM | INT**

**LEVEL=NOMINAL | INTERVAL**

Опція **LEVEL** оператора **INPUT** задає тип для всіх змінних, що вказані в поточному операторі **INPUT**.

Можна задати наступні значення:

- 1) **LEVEL=NOM** – інтерпретує всі змінні як номінальні.
- 2) **LEVEL= INT** – інтерпретує всі змінні як інтервальні.

За замовчуванням **LEVEL=NOM**.

### Оператор **OUTPUT**.

#### **Синтаксис оператора:**

**OUTPUT** < options > ;

Даний оператор використовується для налаштування опцій при виводі результатів роботи процедури **HPBNET**.

**FIT= назва файлу**

Такий запис оператора **FIT** дозволяє здійснити вивід у файл описової статистики побудованої мережі. Наприклад, кількість вершин та зв'язків, значення байєсівського інформаційного критерію.

**NETWORK= назва файлу**

Вивід у файл даних щодо структури мережі та значень ймовірностей станів вершин.

**PARAMETER= назва файлу**

Вивід у файл значень параметрів фінальної моделі.

**PRED= назва файлу**

Вивід у файл прогнозних значень ймовірностей для вхідного набору даних.

**VALIDINFO= назва файлу**

Вивід у файл результатів для валідаційного набору даних.

**VARINFO= назва файлу**

Вивід у файл інформації щодо кожної змінної, а саме: тип, кількість рівнів номінальної змінної, мінімальне та максимальне значення інтервальної змінної, кількість пропусків по змінній, кількість рівнів у змінній, що використовується для побудови топології мережі.

**VARLEVEL= назва файлу**

Вивід у файл детальної інформації щодо кожного рівня кожної змінної.

**VARORDER= назва файлу**

Вивід у файл списку змінних та їхнього порядку для процедури.

**VARSELECT= назва файлу**

Вивід у файл результатів вибору змінних.

**Оператор PARTITION.**

**Синтаксис оператора:**

**PARTITION ROLEVAR=variable (TRAIN="value" | VALIDATE="value");**

або

**PARTITION FRACTION (VALIDATE=number | SEED=integer );**

Оператор **PARTITION** вказує, яким чином розподіляти вхідну вибірку на навчальні та валідаційні набори даних. У процедурі **HPBNET** реалізовано два підходи розподілу вибірки: через **ROLEVAR** або **FRACTION**.

1) Перший підхід через опцію **ROLEVAR** має наступний синтаксис:

**PARTITION ROLEVAR=variable (TRAIN="value" | VALIDATE="value");**

У даному випадку вказується ім'я номінальної змінної на основі значень якої записи відносяться до навчального та валідаційного наборів даних відповідно. Опція **TRAIN="значення"** вказує, які саме записи будуть віднесені до навчальної вибірки, а **VALIDATE="значення"** – до валідаційної. За замовчуванням **TRAIN="1"**, а **VALIDATE="0"**.

2) Другий підхід використовує опцію **FRACTION**, синтаксис якої має наступний вигляд:

**PARTITION FRACTION (VALIDATE=number | SEED=integer );**

У даному випадку опція **VALIDATE=number**, вказує, яка частина вхідних даних буде використана для створення валідаційного набору даних. За замовчуванням **VALIDATE=0,33**. В якості ключа генерації при виборі випадкових



спостережень з вхідного набору даних використовується значення опції **SEED=** ціле число. За замовчуванням **SEED=12345**.

### Оператор **PERFORMANCE**.

**Синтаксис оператора:**

**PERFORMANCE** < options > ;

Оператор **PERFORMANCE** призначений для налаштування паралельних та розподілених обчислень. З особливостями його використання можна детально ознайомитися в третьому розділі технічної документації SAS Enterprise Miner 13.2 High-Performance Procedures, розміщеному на сайті компанії SAS Institute [219].

### Оператор **TARGET**.

**Синтаксис оператора:**

**TARGET** variable </ORDER=order>;

Синтаксис оператора **TARGET** передбачає запис імені змінної, яка використовується процедурою **HPBNET** для прогнозування. Ця змінна повинна бути номінального типу. Для опції **ORDER** можна задати значення **ASC**, **DESC**, **FMTASC**, **FMTDESC**.

1) **ORDER=ASC** – значення рівнів цільової змінної у порядку збільшення на основі реальних значень вхідного набору даних.

2) **ORDER=DESC** – значення рівнів цільової змінної у порядку зменшення на основі реальних значень вхідного набору даних.

3) **ORDER=FMTASC** – значення рівнів цільової змінної у порядку збільшення на основі значень після застосування формату.

4) **ORDER=FMTDESC** – значення рівнів цільової змінної у порядку зменшення на основі значень після застосування формату.

За замовчуванням **ORDER=DESC**.

## Реалізація математичних підходів у HPBNET

### Тести на незалежність

Для вибору змінних та побудови структури мережі використовуються тести на незалежність між двома змінними та умовні тести на незалежність за деякими іншими змінним. Процедура **HPBNET** підтримує тести на незалежність із використанням статистик  $\chi$ -квадрат,  $G$ -статистики та нормалізованого значення взаємної інформації або комбінацій цих статистик.

Будемо вважати, що є номінальні змінні  $X$  та  $Y$  (можливо, вони спочатку були інтервальними, але стали номінальними після відповідних перетворень). Змінна  $X$  має  $r$  рівнів (унікальних значень), а  $Y$  має  $c$  рівнів. У такому випадку формула для обчислення статистики  $\chi$ -квадрат має вигляд:

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - E_{ij})^2}{E_{ij}}, \quad (9.1)$$

де  $O_{ij}$  – частота появи значення з таблиці контингенції, а  $E_{ij}$  – очікувана частота появи значення.

Ступінь свободи тесту дорівнює  $(r-1) \times (c-1)$ . Якщо  $p$ -значення статистики тесту перебільшує заданий рівень, то змінні вважаються незалежними.

Аналогічно обчислюється **G-статистика за формулою:**

$$G^2 = 2 \cdot \sum_{i=1}^r \sum_{j=1}^c O_{ij} \cdot \ln \left( \frac{O_{ij}}{E_{ij}} \right), \quad (9.2)$$

де  $O_{ij}$  – емпірична частота появи, а  $E_{ij}$  – очікувана частота, кількість ступенів свободи тесту  $(r-1) \times (c-1)$ .

Якщо  $p$ -значення статистики тесту перебільшує заданий рівень, то змінні вважаються незалежними.

Значення взаємної інформації між змінними  $X$  та  $Y$  обчислюється за формулою:

$$I(X, Y) = \sum_{x=1}^r \sum_{y=1}^c \left( p(x, y) \cdot \ln \left( \frac{p(x, y)}{p(x) \cdot p(y)} \right) \right), \quad (9.3)$$

де  $p(x, y) = \frac{O_{xy}}{N}$  – це функція сумісного розподілу змінних  $X$  та  $Y$ ,  $p(x) = \frac{\sum_{y=1}^c O_{xy}}{N}$

та  $p(y) = \frac{\sum_{x=1}^r O_{xy}}{N}$  – значення розподілу маргінальних ймовірностей змінних  $X$  та  $Y$  відповідно, а  $N$  це загальна кількість спостережень в навчальному наборі даних.

Нормалізоване значення взаємної інформації має область допустимих значень від 0 до 1 та обчислюється за формулою:

$$NI(X, Y) = \sqrt{1 - \exp(-2 \cdot I(X, Y))}, \quad (9.4)$$

де  $I(X, Y)$  – значення взаємної інформації між змінними  $X$  та  $Y$ .

Якщо нормалізоване значення взаємної інформації менше за допустимий рівень, то обидві змінні вважаються незалежними.

Тест на умовну незалежність між двома змінними  $X$  та  $Y$  по заданій множині  $S$  деяких змінних (до яких не належать  $X$  та  $Y$ ), відповідні статистики додаються один до одного для кожного значення комбінації  $S$ , а відповідна кількість ступенів свободи тестів  $\chi$ -квадрат та  $G$ -квадрат дорівнює:

$$(r-1) \times (c-1) \times q, \quad (9.5)$$

де  $q$  – кількість значень комбінацій для  $S$ .

У процедурі **HPBNET** реалізовано можливість налаштування наступних опцій:

- **INDEPTTEST** – вибір методу тесту на незалежність або декількох.
- **ALPHA** – критичне значення тесту на незалежність із використанням статистики  $\chi$ -квадрат або  $G$ -квадрат.
- **MIALPHA** – критичне значення тесту на незалежність із використанням нормалізованого значення взаємної інформації.

### Вибір вхідних змінних

Мережа Байєса представляє собою пару значень  $\langle G, P \rangle$ , де  $G$  – спрямований ациклічний граф, що складається з множини вершин  $U$ , а  $P$  – множина значень розподілу ймовірностей. Сумісний розподіл значень ймовірностей для вершин  $U$  записується як  $p(U) = \prod_{X \in U} p(X|\pi(X))$ , де  $\pi(X)$  – множина батьківських вершин для  $X$ . Вважається, що топологія мережі  $G$  та розподіл ймовірностей  $P$  відповідають один одному.

Для цільової змінної  $T$  коври Маркова задається підмножина вхідних змінних  $MB \subseteq U - \{T\}$ , така що  $T$  є умовно незалежною від кожної вхідної змінної  $X \subseteq U - MB - \{T\}$  по заданій  $MB$ . Вважається що цільовій змінній  $T$  відповідає єдина унікальна структура коври Маркова.

Процедура **HPBNET** підтримує два підходи щодо вибору змінних. Перший підхід із використанням тестів на незалежність між кожною вхідною змінною та цільовою (випадок, коли задана опція **PRESCREENING=1**). Другий підхід із використанням тестів на умовну незалежність між кожною вхідною змінною та цільовою на заданій множині вхідних змінних (випадок, коли задана опція **VARSELECT=1, 2 або 3**).

Процедура **HPBNET** використовує спеціальні структури даних для ефективного обчислення значень таблиці спряженості будь-якої комбінації змінних, а також динамічні підходи щодо генерації кандидатів з метою зменшення кількості хибних кандидатів (комбінацій змінних).

### Застосування HPBNET для побудови топології мережі Байєса

У загальному випадку процедура HPBNET використовує два підходи для побудови топології мережі на основі функції оцінювання та на основі обмежень. Підхід на основі функції оцінювання використовує функціонал, за допомогою якого зрозуміло, наскільки якісна структура отримана на навчальних даних, та прагне побудувати структуру з найкращим значенням функціоналу. Підхід на основі обмежень використовує тести на незалежність для визначення дуг та їх напрямів [220]. Для реалізації підходу побудови топології на основі функції оцінювання в процедурі HPBNET реалізований байєсівський інформаційний критерій (BIC – Bayesian Information Criterion), що обчислюється за формулою:

$$BIC(G, D) = N \cdot \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} p(\pi_{ij}) \cdot p(X_i = v_{ik} | \pi_{ij}) \cdot \ln(p(X_i = v_{ik} | \pi_{ij})) - \frac{M}{2} \cdot \ln(N), \quad (9.6)$$

де  $G$  – топологія мережі,  $D$  – навчальний набір даних,  $N$  – кількість спостережень в  $D$ ,  $n$  – кількість змінних,  $X_i$  –  $i$ -та змінна,  $r_i$  – кількість рівнів змінної  $X_i$ ,  $\pi_{ij}$  –  $j$ -та комбінація батьків змінної  $X_i$ , а  $M = \sum_{i=1}^n (r_i - 1) \times q_i$  – кількість параметрів розподілу ймовірностей.

Процедура HPBNET використовує тести на незалежність для визначення дуг та їх напрямів. Будемо вважати, що наявні три змінних  $X, Y, Z$ . Виконавши тест на

незалежність, було визначено, що є зв'язок між  $X$  та  $Z$ ,  $Y$  та  $Z$ , але немає зв'язку між  $X$  та  $Y$ . Якщо  $X$  умовно не залежить від  $Y$  по заданій підмножині змінних  $S = \{Z\} \cup S', S' \subseteq U - \{X, Y, Z\}$ , напрямок між  $X$  та  $Z$ ,  $Y$  та  $Z$ , позначається як  $X \rightarrow Z$  та  $Y \rightarrow Z$  відповідно. Відмітимо, що використання тестів на незалежність може не показати орієнтації зв'язків між змінними тому, що деякі структури еквівалентні та мають рівні значення тесту на умовну незалежність. Наприклад,  $X \leftarrow Y \leftarrow Z$ ,  $X \rightarrow Y \rightarrow Z$  та  $X \leftarrow Y \rightarrow Z$  належать до еквівалентних класів. У такому випадку процедура **HPBNET** використовує значення функції байєсівського інформаційного критерію для визначення напрямку зв'язків.

Процедура **HPBNET** дозволяє будувати мережі Байєса різних типів: наївну Байєсівську (NB), дереводоповнену (TAN), наївну розширену мережу Байєса (BAN), мережу батько-дитина (PC) та ковдру Маркова (MB). Залежно від обраного типу мережі використовуються різні алгоритми. Наприклад, для випадку **STRUCTURE=TAN** використовується алгоритм побудови максимально широкого дерева, де в якості вагового значення зв'язку між вершинами використовується значення взаємної інформації. При побудові топології використовуються значення **BESTONE** або **BESTSET** для опції **PARENTING**.

Процедура **HPBNET** впорядковує вхідні змінні на основі значення байєсівського інформаційного критерію функції, що обчислюється за формулою:

$$BIC(X, T) = \max(BIC(X \rightarrow T), BIC(T \rightarrow X)), \quad (9.7)$$

де  $X$  – вхідна змінна, а  $T$  – цільова змінна.

$BIC(T \rightarrow X)$  – це значення байєсівського інформаційного критерію функції у випадку, коли  $X$  батько по відношенню до  $T$  (ігноруючи всі інші змінні), а  $BIC(X \rightarrow T)$  – випадок, коли  $X$  нащадок  $T$  (ігноруючи всі інші змінні).

Процедура **HPBNET** в першу чергу визначає батьківські вершини по відношенню до цільової для структур "батько-дитина" та ковдри Маркова. Після чого визначає батьків для вхідних змінних, що мають найбільше значення байєсівського інформаційного критерію функції по відношенню до цільової. Це продовжується, доки є вхідні змінні з великими значеннями функції байєсівського інформаційного критерію. При визначенні батьківських вершин по відношенню до вхідних, зв'язок визначається на основі тестів на незалежність. Після чого робиться орієнтація (спрямування) зв'язків на основі використання тестів на незалежність та функції байєсівського інформаційного критерію. Процедура **HPBNET** використовує функцію байєсівського інформаційного критерію не тільки для орієнтації зв'язків, а й для контролю складності мережі тому, що складна мережа має багато батьків, за що на функцію байєсівського інформаційного критерію накладається штрафне значення.

Значення **BESTONE** та **BESTSET** опції **PARENTING** намагаються визначити локальну оптимальну структуру для кожної вершини. **BESTONE** додає найкращий кандидат-змінну до батьків на кожній ітерації, в той час як **BESTSET** намагається обрати найкращу множину змінних серед кандидатів множин.

Якщо вхідних змінних забагато, то навчання структури може тривати довго тому, що кількість можливих комбінацій змінних зростає експоненціально. Тому, рекомендується робити попередній етап вибору значимих змінних аналізу.

### Навчання параметрів

Навчання параметрів полягає у визначенні розподілу ймовірностей для кожної вершини структури мережі. Для процедури HPBNET розподіл ймовірностей носить дискретний характер тому, що інтервальні змінні розбиті по групах (інтервалах).

За отриманими результатами у вигляді таблиці розподілу ймовірностей скорінгують записи  $(x_1, x_2, \dots, x_{n-1})$  за формулою:

$$\begin{aligned} \arg \max_c p(T = c | x_1, x_2, \dots, x_{n-1}) &= p(x_1, x_2, \dots, x_{n-1} | T = c) \times K = \\ &= \prod_i p(x_i | \pi(X_i)) \times K \end{aligned} \quad (9.8)$$

де  $c$  – рівень цільової змінної,  $\pi(X_i)$  – батьки змінної  $X_i$ ,  $K$  – константа,  $X_n = T$  – цільова змінна.

### Приклади побудови мереж Байєса засобами процедури HPBNET

#### Приклад 1: наївна мережа Байєса для класифікації рослин

У цьому прикладі показаний випадок побудови наївної мережі Байєса на основі стандартного набору даних *Iris* для загальновідомої задачі прогнозування кольору райдужної оболонки ока. Використовуються наступні налаштування: MAXPARENTS=1, PRESCREENING=0 та VARSELECT=0, тобто процедура використовує лише одну батьківську вершину для кожної змінної та аналізує всі вхідні змінні.

**Примітка.** *Iris* – це рід рослин з роду півникових, також зустрічається під назвою ірис. Набуло назву від грецького слова *Iris*, що означає веселка, через велике різноманіття забарвлення квіток рослин цього роду.



Рис. 9.11. Зовнішній вигляд рослин сімейства ірисових

Такий набір даних широко використовується для демонстрації роботи різноманітних статистичних алгоритмів, вирішення задач класифікації. Набір даних *Iris*, що був створений американським ботаніком Е. Андерсоном, складається з даних про 150 екземплярів ірису, по 50 екземплярів на кожний з трьох видів (*species*) – ірис щетинистий (*Iris setosa*), ірис віргинський (*Iris virginica*) та ірис різнокольоровий (*Iris versicolor*).

Для кожного екземпляра вимірювали чотири характеристики (в сантиметрах):

1. довжина чашолистика (*sepal length*);
2. ширина чашолистика (*sepal width*);
3. довжина пелюстки (*petal length*);
4. ширина пелюстки (*petal width*).

**Код програми:**

```
proc hpbnet data=sashelp.Iris numbin=3 structure=Naive maxparents=1  
prescreening=0 varselect=0;  
target Species;  
input PetalWidth PetalLength SepalLength SepalWidth / level=INT;  
output network=network;  
run;
```

**Код програми для виводу структури в SAS:**

```
proc print data=network noobs label;  
var _parentnode_ _childnode_;  
where _type_="STRUCTURE";  
run;
```

**Результати виконання програми**

```
Parent  
Node Child Node  
  
Species PetalLength  
Species PetalWidth  
Species SepalLength  
Species SepalWidth
```

Структура побудованої мережі Байєса представлена на рис. 9.12.

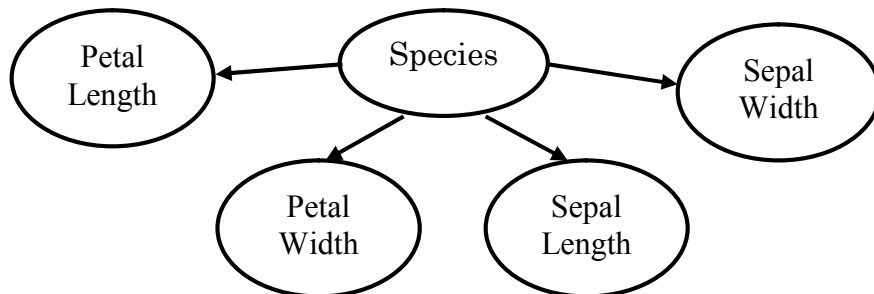


Рис. 9.12. Структура наївної мережі Байєса, побудована на основі набору даних *Iris*.

Як видно з рис. 9.12, *Species* є батьком для *PetalWidth*, *PetalLength*, *SepalLength* та *SepalWidth*.

**Приклад 2: дереводоповнена наївна мережа Байеса для класифікації типу рослини**

У даному прикладі також використано набір даних Iris. Налаштування процедури наступні: STRUCTURE=TAN – структура мережі Байеса у вигляді дереводоповненої наївної мережі, PRESCREENING=0, VARSELECT=0.

**Код програми:**

```
proc hpbnet data=sashelp.Iris numbin=3 structure=TAN
prescreening=0 varselect=0;
target Species;
input PetalWidth PetalLength SepalLength SepalWidth/level=INT;
output network=network;
run;

proc print data=network noobs label;
var _parentnode_ _childnode_;
where _type_="STRUCTURE";
run;
```

**Результати виконання програми:**

```
Parent Node Child Node

Species PetalLength
PetalWidth PetalLength
Species PetalWidth
Species SepalLength
PetalLength SepalLength
Species SepalWidth
SepalLength SepalWidth
```

Як можна побачити з отриманих результатів, Species є батьком для PetalWidth, PetalLength, SepalLength та SepalWidth. Окрім цього, PetalWidth є батьком для PetalLength, SepalWidth батько для SepalLength, а PetalWidth для SepalWidth. Структура мережі TAN наведена на рис. 9.13.

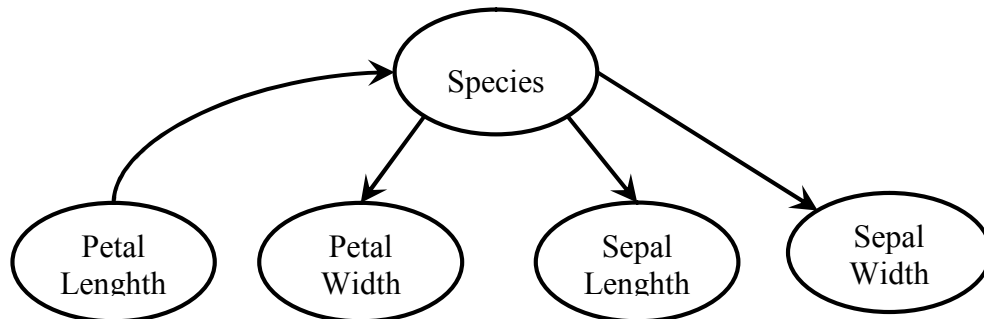


Рис. 9.13. Структура дереворозширеної наївної мережі Байеса, побудована на основі набору даних Iris

**Приклад 3: побудова мережі Байєса "батько-дитина" для класифікації типу рослини**

Для побудови топології мережі Байєса для даних Iris використовується підхід "батько-дитина". Налаштування процедури наступні: STRUCTURE=PC – структура мережі Байєса у вигляді мережі "батько-дитина", PRESCREENING=0, VARSELECT=0.

**Код програми:**

```
proc hpbnet data=sashelp.Iris numbin=3 structure=PC
prescreening=0 varselect=0;
target Species;
input PetalWidth PetalLength SepalLength SepalWidth/level=INT;
output network=network;
run;

proc print data=network noobs label;
var _parentnode_ _childnode_;
where _type_="STRUCTURE";
run;
```

**Результати виконання програми:**

```
Parent Node Child Node

PetalLength Species
Species PetalWidth
Species SepalLength
Species SepalWidth
```

Побудована топологія отриманої структури мережі Байєса представлена на рис. 9.14.

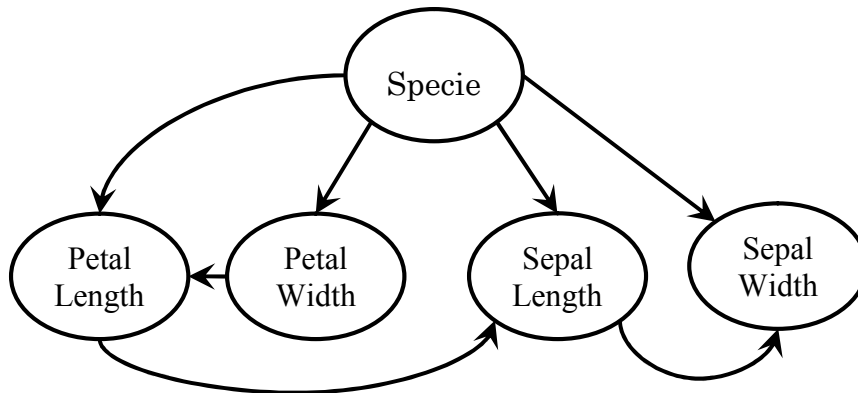


Рис. 9.14. Структура PC мережі Байєса, побудована на основі набору даних Iris



**Приклад 4: побудова мережі Байєса у вигляді ковдри Маркова для визначення кредитоспроможності клієнтів банку**

Для демонстрації можливостей процедури *HPBNET* побудови мереж Байєса у вигляді **ковдри Маркова** (MB – Markov Blanket) використано набір даних HMEQ, що містить інформацію щодо можливості дефолту клієнтів банку.

**Код програми:**

```
proc hpbnet data=sampsio.hmeq indeptest=MI mialpha=0.2 structure=MB
nbin=5 missingint=IMPUTE missingnom=LEVEL;
target Bad;
input Reason Job Delinq Derog Ninq / level=NOM;
input Loan Mortdue Value Yoj Clage Clno Debtinc / level=INT;
output network=network;
run;

proc print data=network noobs label;
var _parentnode_ _childnode_;
where _type_="STRUCTURE";
run;
```

**Результати виконання програми:**

```
Parent Child
Node Node

BAD DELINQ
NINQ DELINQ
BAD DEROG
DELINQ DEROG
NINQ DEROG
```

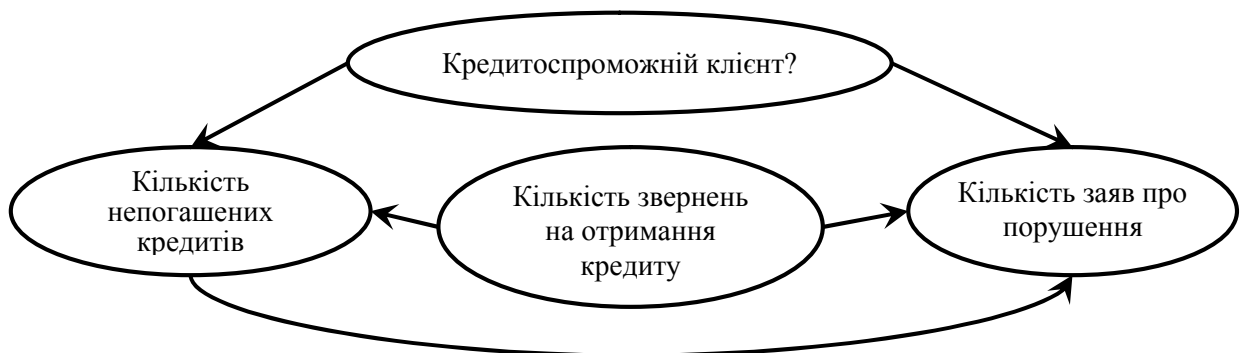


Рис. 9.15. Структура мережі Байєса у вигляді ковдри Маркова для визначення кредитоспроможності клієнта банку

Набір даних **HMEQ** містить інформацію щодо дефолту клієнтів банку, що отримали кредит.

**BAD** – цільова змінна, що може приймати наступні значення:

- BAD=1 – клієнт не повернув кредит;
- BAD=0 – клієнт повернув кредит.

**CLAGE** – скільки місяців тому клієнт отримав кредит, якщо кількість відкритих кредитів "декілька", то розглядається найдавніший.

**CLNO** – кількість відкритих кредитних ліній.

**DEBTINC** – співвідношення заборгованості по кредитах до місячних надходжень клієнта.

**DELINQ** – кількість непогашених кредитів.

**DEROG** – кількість заяв про порушення.

**JOB** – місце роботи (посада, яку посідає клієнт).

**LOAN** – сума кредиту, за яким звернувся клієнт.

**MORTDUE** – кількість відкритих іпотечних кредитів у клієнта.

**NINQ** – кількість звернень від клієнта на отримання кредиту за останній час.

**REASON** – причина звернення за кредитом.

Набуває значення:

- REASON=DebtCon – вирішення фінансових проблем.
- REASON=HomeImp – вирішення проблем, пов'язаних із житлом.

### Приклад 5: побудова доповненої наївної мережі Байєса для визначення кредитоспроможності клієнта банку

У даному прикладі розглянуто вирішення задачі кредитоспроможності на основі вхідного набору даних **HMEQ** із використанням доповненої наївної мережі Байєса.

#### *Код програми:*

```
proc hpbnet data=sampsio.hmeq numbin=10 alpha=0.1
  structure=Naive missingint=IMPUTE missingnom=LEVEL;
  target Bad;
  input Reason Job Delinq Derog Ninq/level=NOM;
  input Loan Mortdue Value Yoj Clage Clno Debtinc/level=INT;
  output network=network;
run;

proc print data=network noobs label;
  var _parentnode_ _childnode_;
  where _type_="STRUCTURE";
run;
```

#### *Результати виконання програми:*

```
Parent Child
Node Node

BAD DELINQ
BAD DEROG
```

DELINQ DEROG  
 BAD JOB  
 BAD NINQ  
 BAD CLAGE  
 BAD CLNO  
 JOB CLNO  
 BAD LOAN  
 BAD MORTDUE  
 JOB MORTDUE

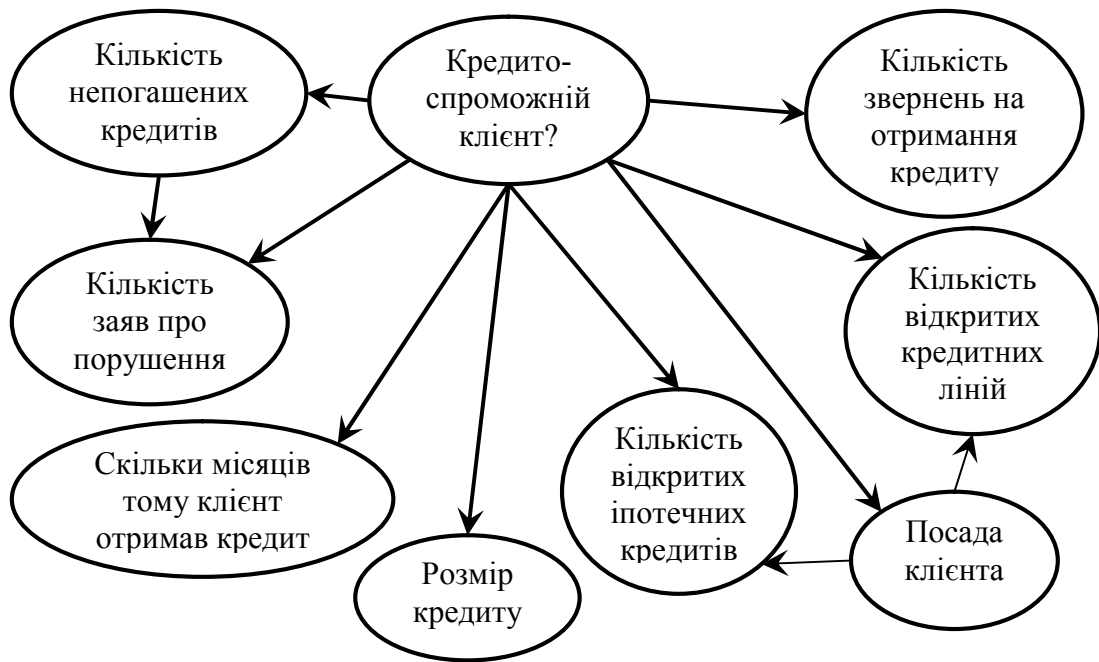


Рис. 9.16. Структура доповненої наївної мережі Байєса для визначення кредитоспроможності клієнта банку

Як можна побачити з отриманих результатів, змінна Bad є батьківською вершиною по відношенню до Delinq, Derog, Job, Ninq, Clage, Clno, Loan та Mortdue. Окрім цього, Delinq є батьком для Derog, а Job є батьком для Mortdue та Clno

**Приклад 6: вибір кращої структури мережі Байєса серед декількох**

У даному прикладі розглянуто вибір кращої структури мережі Байєса серед декількох, а саме: наївної мережі (STRUCTURE=NAIVE), дереводоповненої (STRUCTURE=TAN), "батько-дитина" (STRUCTURE=PC) та ковдри Маркова (STRUCTURE=MB). Для випадків з попереднім вибором змінних та без нього.

В якості вхідних даних використовується набір даних DMAGESR, розроблений на основі даних одного з німецьких банків і який є модельним прикладом задачі. При цьому 30% вибірки використовується для валідації.

**Код програми:**

```
proc hpbnet data=sampsio.dmagecr numbin=10 alpha=0.05
structure=Naive TAN PC MB varselect=0 1 bestmodel;
target Good_Bad;
input Checking History Purpose Savings Employed Installp Marital Coapp
Resident Property Other Housing Existcr Job Depends Telephon
Foreign/level=NOM;
input Age Amount Duration/level=INT;
partition FRACTION (VALIDATE=0.3);
output network=network fit=fit validinfo=validinfo;
run;
```

*Для виводу статистик побудованих моделей* використовується наступний *код програми:*

```
proc print data=fit noobs label;
run;
```

Результати виконання програми:

Number of Observations for Training	Sum of Observations for Validation Ignored	Number of Observations for Training	Sum of Observations for Validation Ignored	Number of Observations for Training	Sum of Observations for Validation Ignored
742	258	0	742	258	0

Maximum Number of Nodes	Number of Links	Average Degree	Parents in Network	Number of Parameters	Score
15	15	2	2	129	-13179.18

Як видно з результатів роботи програми, для навчання використовувалося 742 спостереження, а для валідації 258. При цьому кінцева топологія мережі містить 15 вершин, 15 зв'язків між вершинами, кількість параметрів моделі дорівнює 129.

Набір даних **DMAGECR** розроблений на основі даних одного з німецьких банків, що займається кредитуванням. Опис змінних набору даних:

**CHECKING** – стан чекового рахунку клієнта.

Може мати значення:

- 1 – від'ємний баланс;
- 2 – менше двох сотень доларів;
- 3 – більше двох сотень доларів;
- 4 – немає чекового рахунку.

**DURATION** – тривалість співпраці з клієнтом, місяців.

**HISTORY** – кредитна історія.

Може мати значення:

- 0 – немає активних кредитів;
- 1 – отримані раніше кредити належним чином сплачені;
- 2 – є відкриті кредити, що належним чином сплачуються;
- 3 – є відкриті кредити, за якими є просрочка за платежами;
- 4 – є несплачені або відкриті кредити в іншому банку.

**PURPOSE** – мета кредиту.

Може мати значення:

- 0 – придбання нового автомобіля;
- 1 – придбання автомобіля, що був у вжитку.
- 2 – придбання меблів або спорядження.
- 3 – придбання телевізійної техніки;
- 4 – придбання побутової техніки;
- 5 – ремонт;
- 6 – навчання;
- 7 – відпустка;
- 8 – навчання на другій освіті / підвищення кваліфікації;
- 9 – розвиток бізнесу;
- X – інше.

**AMOUNT** – сума кредиту.

**SAVINGS** – ощадний рахунок.

Може мати значення:

- 1 – менше ста доларів;
- 2 – від 100 до 500 доларів;
- 3 – від 500 до 1000 доларів;
- 4 – більше 1000 доларів;
- 5 – немає ощадного рахунку.

**EMPLOYED** – стаж роботи на теперешньому місці роботи.

Може мати значення:

- 1 – безробітний;
- 2 – менше одного року;
- 3 – від одного до чотирьох років;
- 4 – від чотирьох до семи років;
- 5 – більше семи років.

**INSTALLP** – щомісячні платежі за кредитом по відношенню до місячного доходу клієнта.

**MARITAL** – сімейний стан клієнта.

Може мати значення:

- 1 – чоловік, розлучений;
- 2 – жінка, розлучена або заміжня;
- 3 – чоловік, холостяк;
- 4 – чоловік, одружений;
- 5 – жінка, самотня.

**COAPP** – наявність поручителів.

Може мати значення:

- 1 – немає;

- 2 – є спів заявник;
- 3 – є поручитель.

**RESIDENT** – дата, з якої резидент проживає за вказаною адресою.

**PROPERTY** – наявність власного майна.

Може мати значення:

- 1 – нерухомість;
- 2 – якщо не 1 пункт: є страховий поліс життя / пай на будинок;
- 3 – якщо не 1 і 2 пункти: наявність автомобіля;
- 4 – немає власного майна.

**AGE** – вік, років.

**OTHER** – наявність інших розстрочок.

Може мати значення:

- 1 – банківська;
- 2 – магазину;
- 3 – немає.

**HOUSING** – умови проживання.

Може мати значення:

- 1 – орендоване житло;
- 2 – власне житло;
- 3 – соціальне житло.

**EXISTCR** – кількість непогашених кредитів у даному банку.

**JOB** – професія.

Може мати значення:

- 1 – безробітний, немає ніякої фахової спеціальності;
- 2 – некваліфікований працівник, без офіційного працевлаштування;
- 3 – кваліфікований працівник, з офіційним працевлаштуванням;
- 4 – менеджер / підприємець / висококваліфікований працівник / військовослужбовець.

**DEPENDS** – кількість утриманців.

**TELEPHON** – наявність телефону.

Може мати значення:

- 1 – не вказаний;
- 2 – вказаний.

**FOREIGN** – іноземець.

Може мати значення:

- 1 – так;
- 2 – ні.

**GOOD\_BAD** – цільова змінна.

Може мати значення:

- GOOD – надійний клієнт;
- BAD – ненадійний клієнт.

Для *виводу* отриманої *структури мережі Байєса за методом "батько-дитина"* використовується наступний код програми:

```
proc print data=network noobs label;  
var _parentnode_ _childnode_;  
where _type_="STRUCTURE";
```

run;

Результат:  
Parent Child  
Node Node

checking good\_bad  
good\_bad employed  
good\_bad foreign  
good\_bad history  
good\_bad housing  
good\_bad marital  
good\_bad other  
good\_bad property  
good\_bad purpose  
good\_bad savings  
good\_bad telephon  
foreign telephon  
good\_bad age  
good\_bad amount  
good\_bad duration

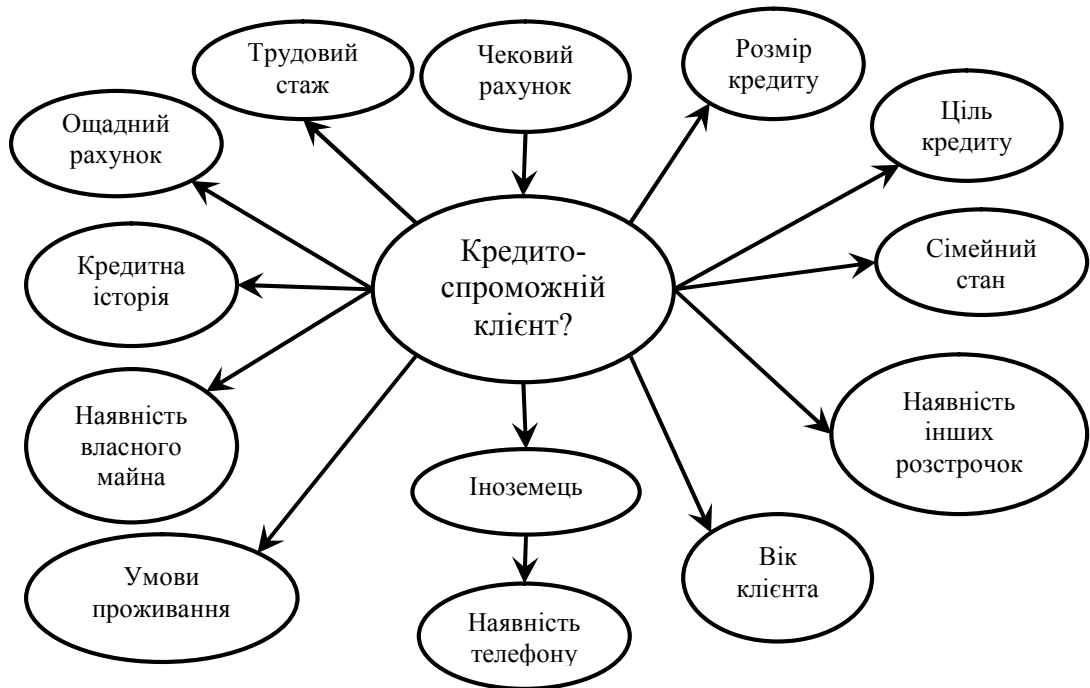


Рис. 9.17. Структура мережі Байєса побудована за методом "батько-дитина" для визначення кредитоспроможності клієнта

**Код програми** для **виводу статистик порівняння моделей** на валідаційному наборі даних:

```
proc print data=validinfo noobs label;
```

```
var _bestmodel_ _miserrors_ _structure_ _varselect_ _maxparents_ ;
run;
```

**Результати роботи програми:**

```

      Input
      Input Parameter:
      Input Parameter: Maximum
Best Misclassification Parameter: Variable Number of
Model Errors   Structure Selection Parents

YES  68  PC   0  2
     68  PC   0  3
     68  PC   0  4
     68  PC   0  5
     72  TAN  0  2
     74  MB   3  2
     74  MB   3  3
     74  MB   3  4
     74  MB   3  5
     74  PC   1  2
     74  PC   1  3
     74  PC   1  4
     74  PC   1  5
     76  PC   0  1
     76  NAIVE 0  1
     76  NAIVE 0  2
     76  NAIVE 0  3
     76  NAIVE 0  4
     76  NAIVE 0  5
     85  MB   3  1
     85  PC   1  1
     85  NAIVE 1  1
     85  NAIVE 1  2
     85  NAIVE 1  3
     85  NAIVE 1  4
     85  NAIVE 1  5
     91  TAN  1  2

```

На підставі отриманих результатів можна зробити наступні висновки:

1. Мережа Байєса, побудована за методом "батько-дитина" на валідаційному наборі даних неправильно класифікувала 68 спостережень з 258 з опціями: MAXPARENTS, що більше або дорівнює 2, VARSELECT=0.
2. Мережа Байєса побудована TAN методом дає неправильну класифікацію в 70 з 258 валідаційних спостережень.
3. Ковдра Маркова дає неправильну класифікацію в 74 з 258 валідаційних спостережень з опцією MAXPARENTS, що більше або дорівнює 2.
4. Наївна мережа Байєса дає неправильну класифікацію в 76 з 258 валідаційних спостережень.



#### 9.4. Приклади побудови систем підтримки прийняття рішень із використанням мереж Байєса

Одним із прикладів системи підтримки прийняття рішень з використанням мереж Байєса є комп'ютерна система, що спрямована на розв'язання таких задач інтелектуального аналізу даних, як аналіз причинно-наслідкових зв'язків, класифікація, кластеризація, прогнозування та візуалізація. Розв'язання цих задач здійснюється в основному без залучення експертів – за навчальними даними. Залучення експертів можливе лише при аналізі слабоструктурованих проблем. Головна мета розробленої системи підтримки прийняття рішень - надання допомоги особам, що приймають рішення.

Структурна схема розробленої системи підтримки прийняття рішень BNetMaster зображена на рис. 9.18.

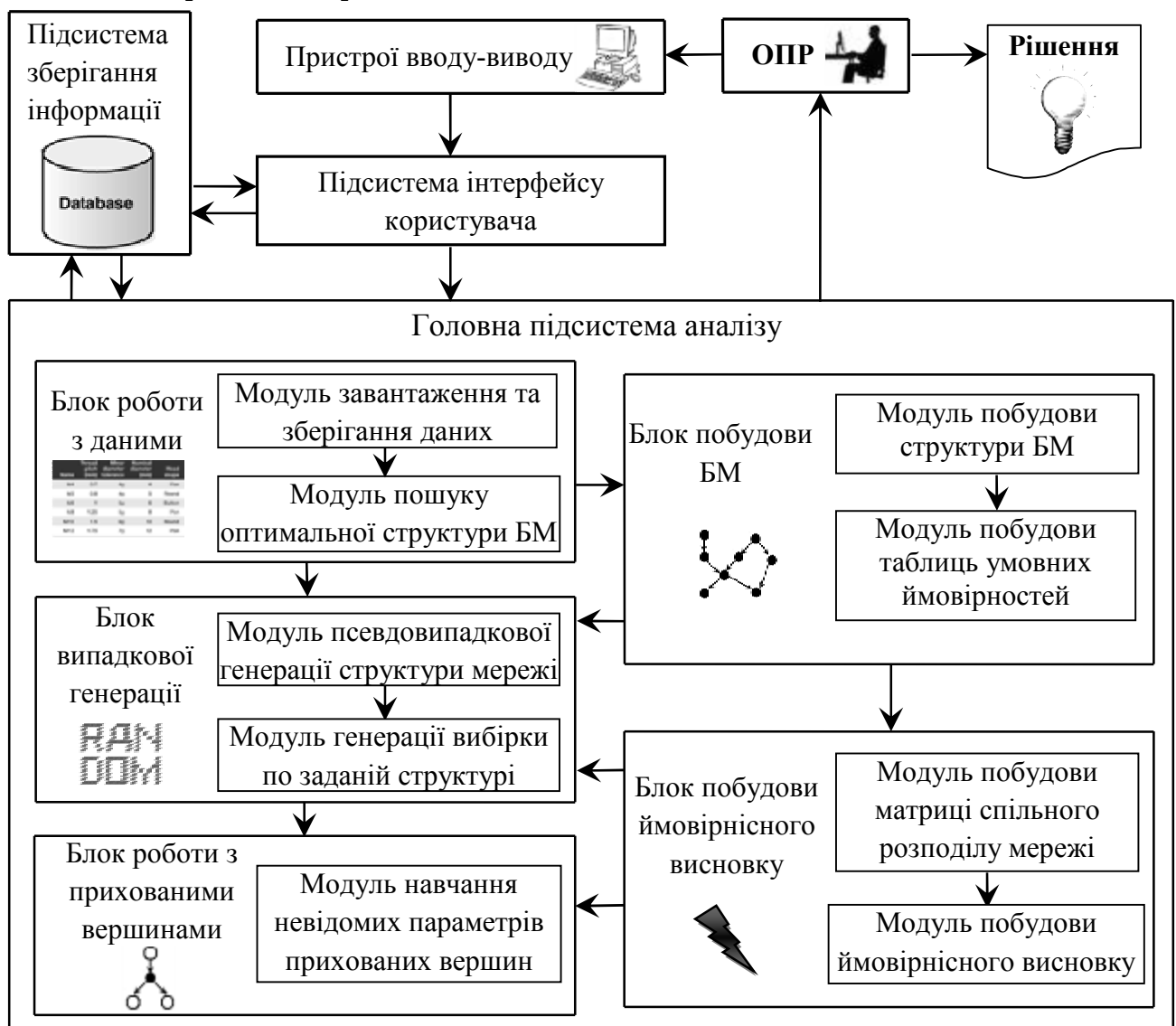


Рис. 9.18. Архітектура системи підтримки прийняття рішень BNetMaster

Система складається з трьох основних підсистем і має блочно-модульну архітектуру.

**Пристрої вводу-виводу** дозволяють користувачеві завантажувати дані для побудови мережі Байєса та для навчання параметрів прихованих вершин, а також зберігати результати роботи в зовнішні файли.

**Підсистема інтерфейсу користувача** пов'язує користувача, внутрішні елементи СППР, зовнішні запам'ятовуючі пристрої та пристрої вводу-виводу. Інтерфейс надає можливість користувачеві завантажувати дані, вводити команди і запити в систему, отримувати графічне представлення даних і результатів, зберігати отримані результати.

**Підсистема зберігання інформації** складається із загальної бази даних, бази моделей та бази знань, які призначені для накопичення навчальних даних, готових моделей на основі мереж Байєса.

**Головна підсистема аналізу** система призначена для аналізу даних на основі БМ, тобто побудова БМ та ймовірнісного висновку, генерація вибірки по заданій структурі мережі та навчання параметрів прихованих вершин. Головна підсистема аналізу складається з п'яти блоків: (1) блок роботи з даними; (2) блок побудови БМ; (3) блок побудови ймовірнісного висновку; (4) блок випадкової генерації; (5) блок роботи з прихованими вершинами. Дані до цієї підсистеми надходять від підсистеми зберігання інформації по шині даних.

**Блок роботи з даними** призначений для пошуку оптимальної структури мережі Байєса, завантаження даних для її побудови, а також зберігання даних, отриманих як результат генерування вибірки. Даний блок складається з наступних модулів: (1) модуль завантаження та зберігання даних; (2) модуль пошуку оптимальної структури БМ.

**Модуль завантаження та зберігання даних** призначений для завантаження даних з файлу та вивантаження згенерованих даних.

**Модуль пошуку оптимальної структури мережі Байєса** використовує евристичний алгоритм побудови мережі для знаходження топології мережі з можливістю вибору способу обчислення взаємозв'язку між вершинами серед наступних: значення взаємної інформації, коефіцієнти Пірсона, Крамера, Чупрова і лямбда Гудмана.

**Блок побудови байєсових мереж** дає можливість будувати мережі «вручну», використовуючи графічний інтерфейс. Блок складається з таких модулів: (1) модуль побудови структури БМ; (2) модуль побудови таблиць умовних ймовірностей.

**Модуль побудови структури мережі Байєса** реалізує інструменти для побудови БМ «вручну» і дає можливість будувати та видаляти вершини, зв'язки між ними, відображати чи приховувати імена вершин та очищати поле побудови.

**Модуль побудови таблиць умовних ймовірностей** дозволяє редагувати ім'я вершини, змінювати стани вершин, додавати, видаляти, перейменовувати їх та заповнювати таблиці умовних ймовірностей.

**Блок побудови ймовірнісного висновку** призначений для побудови ймовірнісного висновку по структурі мережі Байєса, що надходить з блоку побудови її структури. Блок складається з модуля побудови матриці спільного розподілу та модуля побудови ймовірнісного висновку.

**Модуль побудови матриці спільного розподілу** призначений для побудови матриці емпіричних значень сумісного розподілу ймовірностей всієї мережі Байєса.

**Модуль побудови ймовірнісного висновку** здійснює ймовірнісний висновок в мережі на основі матриці емпіричних значень спільного розподілу ймовірностей.

**Блок випадкової генерації** призначений для псевдовипадкової генерації структури мережі за заданими параметрами та формування вибірки за заданою структурою. До складу блоку входять такі модулі:

- модуль псевдовипадкової генерації структури мережі;
- модуль генерації вибірки по заданій структурі.

**Модуль псевдовипадкової генерації структури мережі.** Для того, щоб виключити висячі вершини, першим етапом було поєднання всіх вершин у ланцюг. При цьому напрямок зв'язку обирається випадковим чином. Кількість зв'язків побудованих таким чином  $N - 1$ .  $M - N + 1$  зв'язків, що залишилися, будуються шляхом вибору випадковим чином батьківської та дочірньої вершин. При додаванні зв'язку здійснюється перевірка на циклічність. Якщо новий зв'язок утворює цикл, то він відкидається і обираються нові батьківська і дочірня вершини.

Таблиці умовних ймовірностей задаються шляхом заповнення їх випадковими числами в діапазоні від 0 до 1 із наступною нормалізацією (зведення суми ймовірностей по рядку до 1).

**Модуль генерації вибірки по заданій структурі.** Генерування вибірки відбувається наступним чином: спочатку робиться ймовірнісний висновок у мережі без інстанційованих вершин. Далі по черзі інстанціюються стани кожної вершини, які утворюють необхідну кількість записів у вибірці.

**Блок роботи з прихованими вершинами** складається з одного модуля навчання невідомих параметрів прихованих вершин і призначений для ітераційного знаходження параметрів мережі з використанням ЕМ-алгоритму за критерієм логарифмічної функції правдоподібності.

Іншим прикладом побудови системи підтримки прийняття рішень є **задача прогнозування продажів на наступні періоди**. Нехай компанія з іноземним капіталом збирає статистику щодо рівня вторинних продажів в кожному тижні і місяці по окремих регіонах, дистриб'юторах та видах товарів. Необхідно встановити ймовірність зростання продажів і спрогнозувати обсяг продажів у наступні періоди, а також оцінити, наскільки охоплений конкретний регіон продажами, чи добре працює дистриб'ютор в даному регіоні.

Задача описується такими змінними: назва регіону, дистриб'ютора або ключового клієнта, що працює в регіоні, тип (категорія) продукту, який продається, результат прогнозу – збільшення чи зменшення продажів у наступному місяці. Іноземна компанія не має власного виробництва в Україні, а тому товар завозиться з іншої країни. Тому якісне прогнозування обсягів продажів у наступні періоди дозволить правильно організувати логістику, тобто вчасне завезення товару у необхідних обсягах задля того, щоб увесь товар реалізувався і не було необхідності його десь зберігати.

У рамках загальної інформаційної системи, що створюється, необхідно виконати проектування і реалізацію бази даних для підприємства як основного сховища для зберігання і обробки статистичної інформації стосовно кількісних і вартісних показників бізнес-діяльності. Основні типи операцій, які виконуються під час бізнес-діяльності і мають знайти відображення в автоматизованій системі – це

операції щодо закупівлі товару у постачальників, реалізації товару клієнтам, ведення обліку товарів, статистичні підрахунки і оформлення звітної документації (у тому числі у вигляді товарних чеків і т. ін., а також у вигляді звітів за певний період стосовно результатів фінансової діяльності підприємства). Інформація, яка використовується під час діяльності, має бути збережена з можливістю її подальшого використання в рамках аналізу чи видачі певних даних за звітний період [219].

Статистичні дані стосовно рівня продажів збираються в одиницях товару (контейнерах, упаковках) і грошових одиницях (доларах США та гривнях). Для об'єктивності, враховуючи коливання курсу національної валюти в Україні, слід оцінювати рівні продажів кількістю контейнерів або в доларах США. Вхідні дані, які використовуються для аналізу, приводяться до вигляду, представленого у табл. 9.6.

Таблиця 9.6

Приклад структуризації вхідних даних

Тип продукту	Місяць	Категорія продукту	Регіон	Назва дистриб'ютора	Обсяг продажів, ящиків	Обсяг продажів, долл США	Прогноз на наступний місяць
Тип 1	Січень	Бренд 1	Київ	Дистриб'ютор 1	...	...	Зростання (1)
Тип 2	Лютий	Бренд 2	Черкаси	Дистриб'ютор 2	...	...	Падіння (0)
...	...	...	...	...	...	...	...

На основі статистичних даних за допомогою інтегрованого підходу відбираємо суттєві змінні, які застосовуємо для побудови моделі на основі мережі Байєса. Структура побудованої мережі Байєса показана на рис. 9.19.

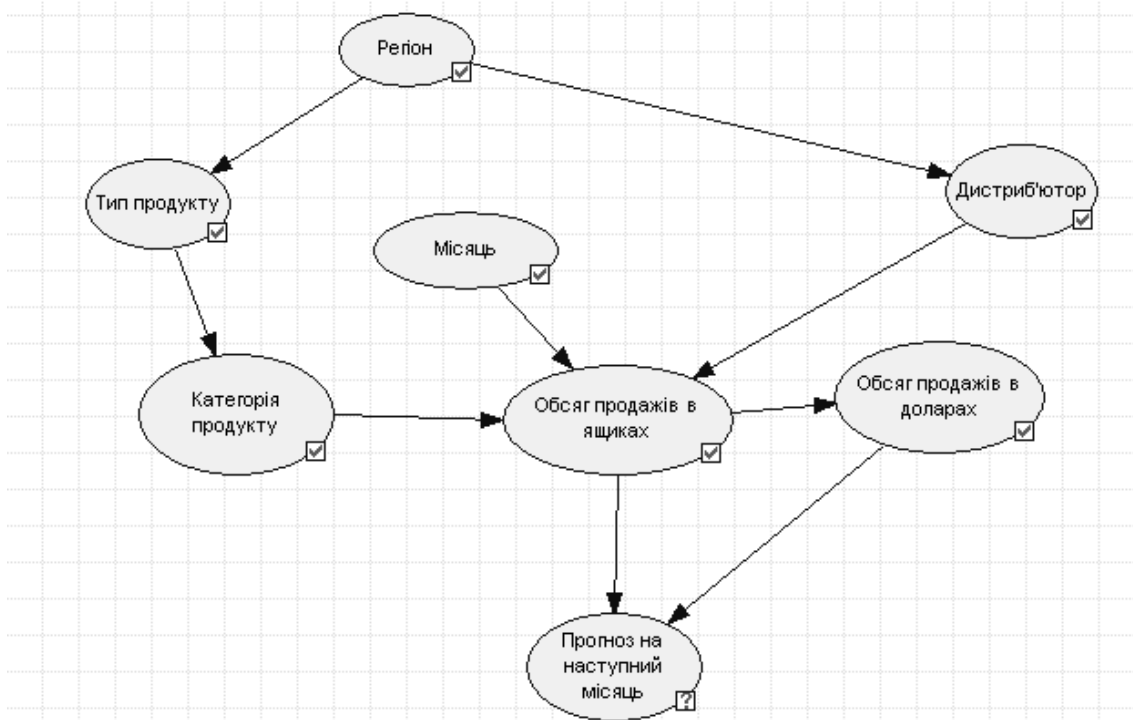


Рис. 9.19. Структура мережі Байєса для задачі аналізу продажів компанії

Побудована структура мережі Байєса дозволяє визначити ймовірність зростання продажів у наступні періоди на основі вхідних даних.

Очевидно, що для стратегічного планування компанії слід збирати та враховувати статистику продажу товарів разом із широтою охоплення конкретним брендом регіонів України, щоб надалі ретельніше планувати маркетингову стратегію, враховуючи чисельність населення в регіоні, співвідношення чоловічого та жіночого населення, середній рівень заробітної плати у регіоні тощо. Це дозволило б визначити зв'язки і впливи між цими змінними і орієнтувати продажі та акції на кінцевого споживача.

Основні вимоги до інформаційної технології та інформаційної СППР такі:

- *адекватність* поставленій задачі;
- *своєчасність* задоволення інформаційних та обчислювальних потреб при проведенні аналізу фінансової діяльності;
- *мінімальний час відгуку* на аналітичні запити;
- можливість *зручного представлення* вихідної інформації (наприклад, у табличному та графічному вигляді);
- можливість *внесення коректив* у методика розрахунку та форму відображення кінцевого результату;
- можливість *повторного вирішення* завдання з будь-якої стадії розрахунку;
- можливість роботи із системою *декільком користувачам* у мережі;
- *простота та дружність інтерфейсу* і легкість взаємодії людини-системи.

Взаємини між суб'єктами системи підтримки прийняття рішень – від постановки проблеми до ухвалення рішення – розрізняються в кожній окремо узятій організації. На відміну від інших підприємств та установ, у фінансовому секторі основна увага сконцентрована на структурі і можливостях аналітичного програмного комплексу. Аналітичний програмний комплекс – універсальний інструментальний засіб збору, консолідації, обробки і аналізу великих обсягів інформації. Він здатний на основі наявних даних, наприклад про клієнтів банку, філій, зроблених банківських операціях, послідовності зовнішніх подій тощо, аналізувати поточний стан головного банку і його філій, а також клієнтів банку, оцінювати стан банків-кореспондентів, прогнозувати поведження різних показників, автоматично відслідковувати, що відбувається і чи насуваються критичні події, аналізувати взаємозв'язок подій і процесів, що відбуваються одночасно, а також подій, зміщених у часі, на основі виявлених взаємозв'язків прогнозувати поведження одних показників у залежності від значень інших.

Згідно описаним вище вимогам спроектована інформаційна технологія аналізу фінансових даних всередині банку (оцінювання фінансового стану компаній, окремих позичальників, прогнозування обсягів депозитів тощо). Технологія представляє собою сукупність методів, програмних і технічних засобів, об'єднаних в єдиний технологічний ланцюжок, що забезпечує збір, збереження, редагування, обробку, представлення та розповсюдження інформації. Запропонована інформаційна технологія передбачає спільне використання певним чином модифікованої бази даних клієнтів, програмного комплексу GeNIe 2.0, зручного інтерфейсу для менеджерів, які виконують роботу з клієнтами, та додаткових програмних модулів побудови спеціальних оцінок (наприклад, індексу GINI та інших). Прийняття рішення, наприклад стосовно можливості надання клієнту кредиту, здійснюється менеджером банку на основі отриманого значення ймовірності

повернення кредиту. Всі етапи формальної перевірки документуються і зберігаються в базі даних банку.

Інформаційна технологія реалізована у вигляді інформаційної СППР, архітектура якої представлена на рис. 9.20.

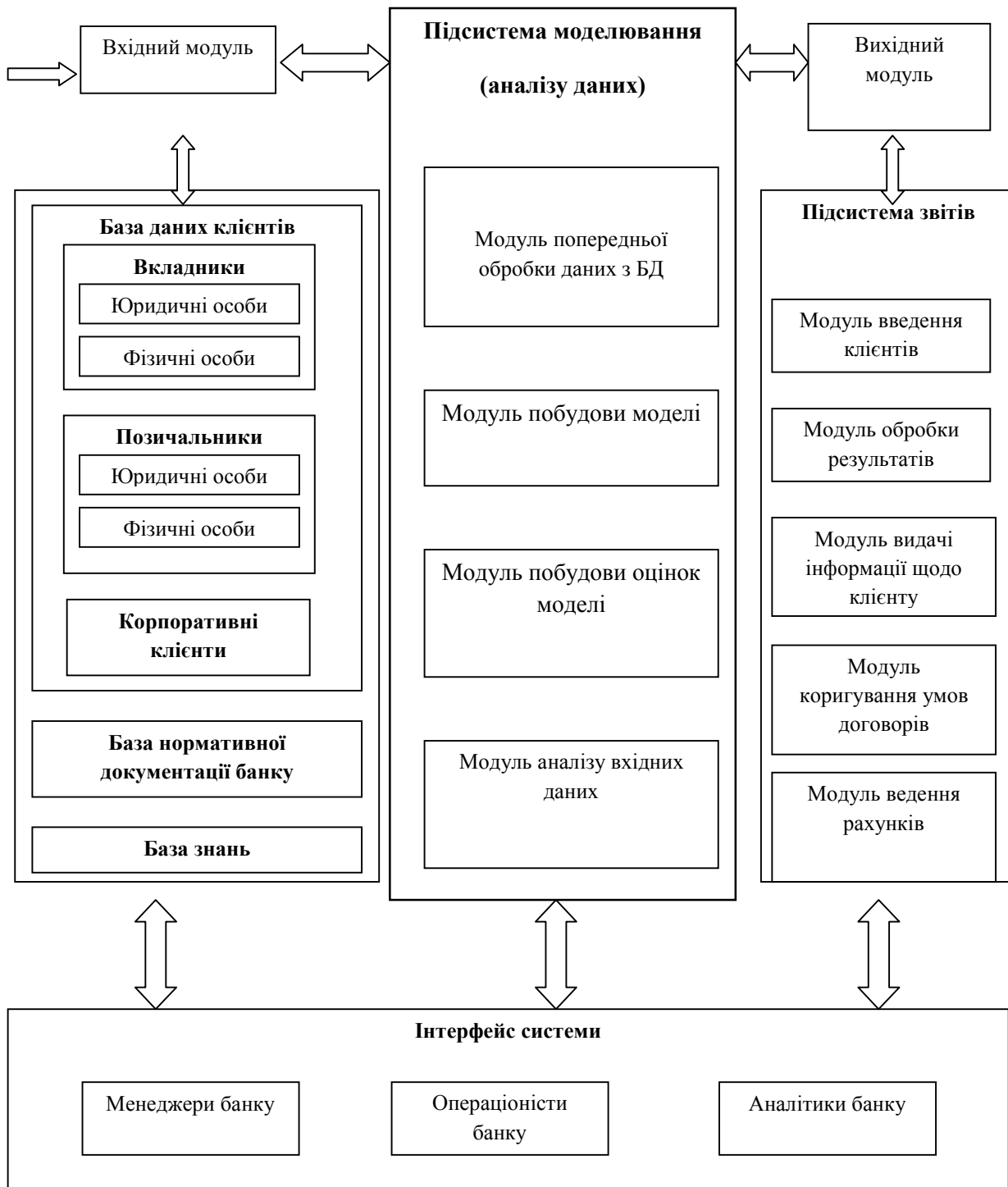


Рис. 9.20. Архітектура інформаційної системи підтримки прийняття рішень у комерційному банку

### Контрольні питання

1. Наведіть приклади використання мереж Байєса у системах підтримки прийняття рішень.
2. Яка процедура в системі SAS 9.4 використовується для побудови топології мережі Байєса?
3. Які опції та оператори процедури побудови мережі Байєса визначають тип мережі?
4. Наведіть код SAS-програми для відображення побудованої структури мережі Байєса.
5. Яким чином відбувається налаштування методу тестування на незалежність змінних? Наведіть відповідний приклад коду програми.
6. Чому дорівнюють значення порогу тесту на незалежність для критеріїв  $\chi$  та  $G$  квадрат?
7. За допомогою яких опцій відбувається обробка пропущених значень? в навчальному наборі даних, при побудові топології мережі Байєса в SAS?
8. Яким чином відбувається перетворення інтервальної змінної до номінального типу? Яка можлива область допустимих значень кількості інтервалів в SAS та яке значення використовується по замовчанню?
9. Які типи структур мереж Байєса підтримуються в SAS? Наведіть приклад відповідного коду програми.
10. Яким чином задається вибір кращої структури мережі Байєса серед декількох побудованих в системі SAS?
11. Надайте приклад та сенс використання операторів PRESCREENING та VARSELECT.
12. Для чого використовуються оператори Code, Freq та ID? Наведіть приклади використання.
13. Наведіть та поясніть опції, що підтримуються оператором Output, при виводі результатів роботи процедури побудови топології мережі Байєса.
14. Для чого використовується оператор Partition? Які підходи для розбиття вибірок реалізовані та в яких випадках ними треба користуватися?
15. Поясніть призначення операторів Performance та Target.
16. Наведіть формулу обчислення статистики  $\chi$ -квадрат та наведіть приклад її використання при побудові мережі Байєса.
17. Наведіть формулу обчислення статистики  $G$ -квадрат та наведіть приклад її використання при побудові мережі Байєса.
18. Наведіть формулу обчислення нормалізованого значення взаємної інформації та наведіть приклад її використання при побудові мережі Байєса.
19. Яким чином відбувається побудова топології мережі Байєса в SAS? Наведіть формулу для обчислення значення байєсівського інформаційного критерію.
20. Яким чином відбувається навчання параметрів мережі Байєса в SAS? Наведіть формулу для обчислення та наведіть пояснення.

## ЗАВДАННЯ ДЛЯ ВИКОНАННЯ ПРАКТИЧНИХ РОБІТ

### ПРАКТИЧНА РОБОТА 1

**Мета:** Ознайомлення з основами теорії байєсівських мереж. Формула Байєса для обчислення значень ймовірностей.

#### Завдання

1. **Ознайомтеся з інформацією**, наведеною в файлах  
02\_Info\_01.doc (основний документ)  
03\_Info\_02.doc (допоміжний, надана більш розширена інформація)  
04\_Info\_03.doc (приділіть увагу задачам 3-5 обчислення ймовірностей)
2. **Напишіть комп'ютерну програму**, що повинна:
  - обчислювати значення ймовірностей станів вершин
  - обчислювати значення спільної ймовірності мережі Байєса

Мову програмування розробник може обирати на власний розсуд. Програма може не будувати графічну структуру мережі Байєса, адже головна мета – коректно запрограмувати обчислення значень ймовірностей за відповідними формулами. Вхідними даними програми є значення умовних ймовірностей вершин.

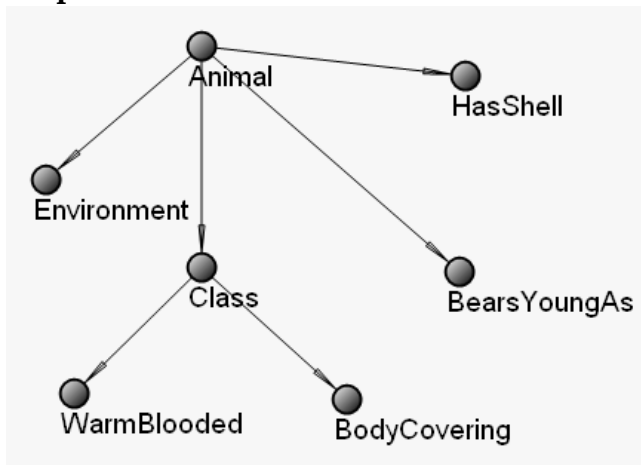
3. **Виконайте обчислення** із використанням написаної програми, використовуючи дані, подані з файлу 05\_data\_for\_net.doc (Якщо виникають складності щодо написання універсальної програми для обчислення значень ймовірностей, то можна написати дві окремі програми для мереж *Animals* та *Asia*).

4. **Результати виконання** роботи необхідно оформити у вигляді звіту, який повинен містити:
  - структуру мережі Байєса;
  - результати роботи програми;
  - код програми для обчислення значень ймовірностей;
  - висновки.

#### Вихідні дані для виконання практичної роботи

##### Мережа *Animals*

#### 1. Структура мережі.





2. Таблиці умовних ймовірностей

Environment

Animal	<i>Air</i>	<i>Land</i>	<i>Water</i>
Monkey	0	1	0
Penguin	0	0,5	0,5
Platypus	0	0	1
Robin	0,5	0,5	0
Turtle	0	0,5	0,5

HasShell

Animal	<i>True</i>	<i>False</i>
Monkey	0	1
Penguin	0	1
Platypus	0	1
Robin	0	1
Turtle	1	0

Animal

<i>Monkey</i>	<i>Penguin</i>	<i>Platypus</i>	<i>Robin</i>	<i>Turtle</i>
0,2	0,2	0,2	0,2	0,2

BearsYoungAs

Animal	<i>Live</i>	<i>Eggs</i>
Monkey	1	0
Penguin	0	1
Platypus	0	1
Robin	0	1
Turtle	0	1

Class

Animal	<i>Bird</i>	<i>Mammal</i>	<i>Reptile</i>
Monkey	0	1	0
Penguin	1	0	0
Platypus	0	1	0
Robin	1	0	0
Turtle	0	0	1

WarmBlooded

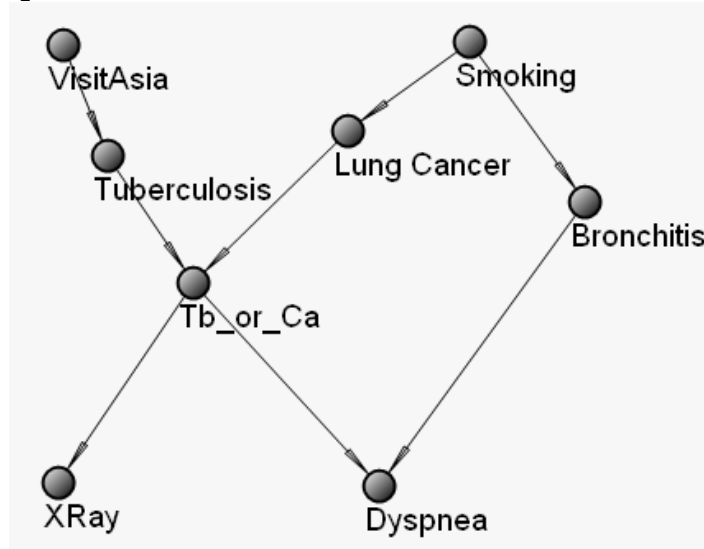
Class	<i>True</i>	<i>False</i>
<i>Bird</i>	1	0
<i>Mammal</i>	1	0
<i>Reptile</i>	0	1

**BodyCovering**

Class	<i>Fur</i>	<i>Feathers</i>	<i>Scales</i>
<i>Bird</i>	0	1	0
<i>Mammal</i>	1	0	0
<i>Reptile</i>	0	0	1

**Мережа Asia**

1. Структура мережі



2. Таблиці умовних ймовірностей

**Tuberculosis**

VisitAsia	<i>Present</i>	<i>Absent</i>
<i>Visit</i>	0,05	0,95
<i>NoVisit</i>	0,01	0,99

**Lung Cancer**

Smoking	<i>Present</i>	<i>Absent</i>
<i>Smoking</i>	0,1	0,9
<i>NoSmoking</i>	0,01	0,99

**XRay**

Tb_or_Ca	<i>abnormal</i>	<i>normal</i>
<i>True</i>	0,98	0,02
<i>False</i>	0,05	0,95

**Smoking**

<i>Smoking</i>	<i>NoSmoking</i>
0,5	0,5

**VisitAsia**

<i>Visit</i>	<i>NoVisit</i>
0,01	0,99

**Tb\_or\_Ca**

<b>Tuberculosis</b>	<b>Lung Cancer</b>	<i>True</i>	<i>False</i>
<i>Present</i>	<i>present</i>	1	0
<i>Present</i>	<i>Absent</i>	1	0
<i>Absent</i>	<i>present</i>	1	0
<i>Absent</i>	<i>Absent</i>	0	1

**Bronchitis**

<b>Smoking</b>	<i>Present</i>	<i>Absent</i>
<i>Smoking</i>	0,6	0,4
<i>NoSmoking</i>	0,3	0,7

**Dyspnea**

<b>Tb_or_Ca</b>	<b>Bronchitis</b>	<i>True</i>	<i>False</i>
<i>True</i>	<i>Present</i>	0,9	0,1
<i>True</i>	<i>Absent</i>	0,7	0,3
<i>False</i>	<i>Present</i>	0,8	0,2
<i>False</i>	<i>Absent</i>	0,1	0,9

**Контрольні запитання:**

1. Що таке мережа Байєса?
2. Мережа Байєса має вигляд спрямованого не спрямованого графа?
3. Наведіть типи структур мереж Байєса.
4. Що таке полі-дерево, чим відрізняється від інших структур?
5. Дайте визначення апіорної, апостеріорної та умовної ймовірностей.
6. Дайте визначення формули Байєса.
7. Охарактеризуйте типи мереж Байєса. Чим вони відрізняються?
8. Поясніть формулу обчислення спільної ймовірності мережі Байєса. Наведіть приклад застосування.
9. Що таке дискретна мережа Байєса? Чим відрізняється від мереж інших типів?

**ПРАКТИЧНА РОБОТА 2**

**Мета:** Ознайомлення з програмою Netica. Побудова ймовірнісного висновку в мережах Байєса – обчислення значень ймовірностей станів вершин.

**Завдання**

1. Ознайомтеся з інформацією, що міститься в файлах **02\_Info\_01.doc** (приділіть увагу прикладу обчислення значень ймовірностей станів вершин за наявності в

мережі інстанційованих вершин) та **03\_Manual\_Netica.doc** (інструкція з експлуатації програмою Netics).

2. **Побудуйте мережу Байєса** із використанням комп'ютерної програми **Netica** за даними, наведеними в **03\_Manual\_Netica.doc** щодо призначеної оцінки кредитоспроможності клієнта.

3. **Завантажте готові приклади** мереж Байєса:

**Chest Clinic** – мережа для діагностування пацієнта (в літературі ця мережа нерідко зустрічається під назвою Asia)

**Car diagnostic** – мережа для визначення ймовірності того, що автомобіль заводиться.

Готові приклади мереж **Chest Clinic** та **Car diagnostic** можна знайти в папці **Data**.

4. Для мережі **Chest Clinic** задайте наступні інстанційовані значення

**Smoking** = smoker

Visit to Asia = visit

та обчисліть значення ймовірностей вершин **X Ray Result** та **Dyspnea** за допомогою програми **Netica** та **вручну** (по аналогії з прикладом наведеним в 02\_Info\_01.doc )

5. **Результати виконання** лабораторної роботи необхідно оформити у вигляді **звіту**, який повинен містити:

– структуру мережі Байєса призначеної оцінки кредитоспроможності клієнта, **Chest Clinic** та **Car diagnostic**;

– для кожної мережі змоделювати 2-3 ситуації наявності інстанційованих значень вершин в мережі.

Результати слід представити у вигляді таблиці. Наприклад, для мережі оцінки кредитоспроможності клієнта, таблиця результатів моделювання може мати вигляд:

№	Інстанційовані вершини	Результат моделювання
1	TypeOfContract = Empl_Full_Time Age = more_36	P(Result = bad) = 6,57% P(Result = Good) = 93,43%
2	ContactPerson = no MaritalStatus = Widowed	P(Result = bad) = 7,79% P(Result = Good) = 92,21%

– Результати обчислення вручну та із використанням програми **для мережі Chest Clinic** представити у вигляді таблиці:

Вершина	Результати Netica	Результати, одержані вручну
X Ray Result = abnormal		
X Ray Result = normal		
Dyspnea = present		
Dyspnea = absent		

– Висновки по роботі.

**Контрольні запитання:**

1. Що таке інстанційвання вершина мережі Байєса?
2. Результати отримані в пункті 4 вручну та із використанням програми Netica відрізняються. Чому?
3. Що таке ймовірнісний висновок в мережі Байєса? Наведіть приклад.

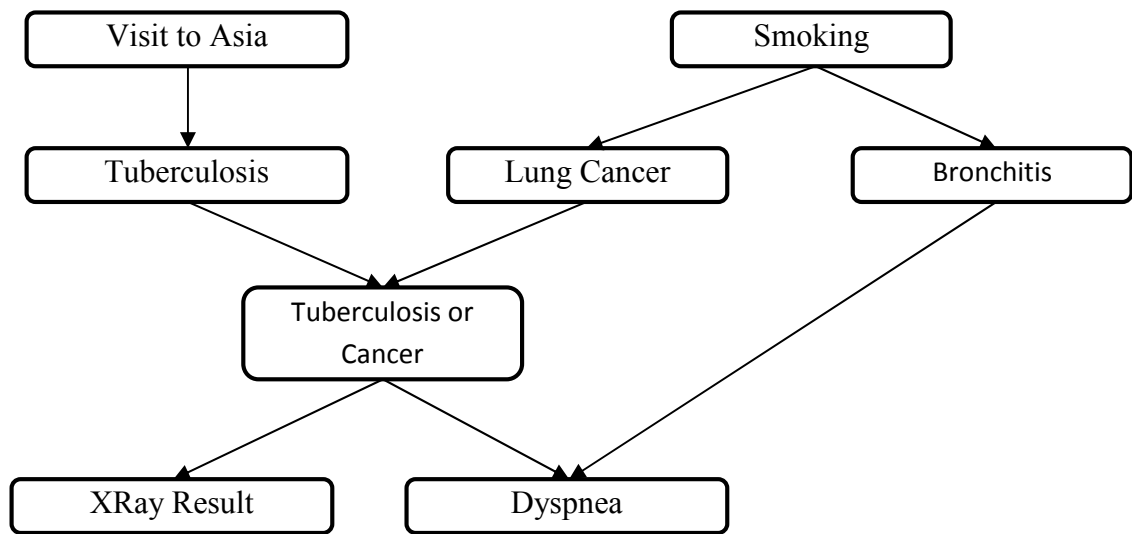
**ПРАКТИЧНА РОБОТА 3**

**Мета:** Навчитися програмувати метод опису мініальною довжиною. Набути навичок апробувати алгоритм на модельній задачі "байєсівська мережа Asia". Ознайомитися з різними методами щодо оцінювання якості побудови мереж Байєса.

**Завдання**

1. **Опрацюйте** теоретичну інформацію: обчислення значення взаємної інформації (ЗВІ) та функції ОМД (Опис Мініальною Довжиною, англ. MDL – Minimum Description Length); алгоритмом евристичного методу побудови байєсовської мережі на основі використання ЗВІ та функції ОМД. Дані матеріали містяться у файлі **02\_LR\_Description\_MDLdoc**.
2. **Реалізуйте програмно** евристичний метод побудови топології мережі Байєса на основі використання ЗВІ і функції ОМД. Мову програмування можна обирати на власний розсуд.
3. **Побудуйте топологію** мережі Asia з використанням експериментальних статистичних даних, які знаходяться в ... \ Data \ Asia.txt.
4. **Обчисліть помилку** між структурою мережі Asia, побудованої розробленим в рамках виконання завдання методом та еталонною структурою. Опис методів оцінювання представлений у файлі **03\_LR\_3\_Assessment\_BN.doc**.

**Обов'язково.** Для обчислення похибки потрібно використовувати формулу структурної різниці. Еталонна структура має вигляд *Asia*:



Еталона структура мережі *Asia*

**Необов'язково.** Реалізувати критерій перехресної ентропії.

5. **Результати виконання** лабораторної роботи необхідно оформити у вигляді звіту, який повинен містити:

- результати обчислювального експерименту у вигляді таблиць:

Таблиця 1

Результати обчислювального експерименту

Час роботи програми	Години: хвилини: секунди: мілісекунди
Загальна кількість моделей, які були проаналізовані запрограмованим методом	Число
Кількість зайвих дуг	Число
Відсутні дуги	Число
Реверсовані дуги	Число
Структурна різниця між побудованою та еталонною структурами мережі <i>Asia</i>	Число

- структуру побудованої мережі *Asia* за допомогою реалізованої програми, час роботи програми;
- таблицю значень взаємної інформації між усіма парасполученнями вершин мережі;
- графік зміни функції ОМД залежно від ітерацій побудови структури мережі;
- значення функції ОМД для випадку, коли всі вершини БС незалежні;
- програмний код алгоритму ОМД.

**Контрольні запитання:**

1. Які існують методи оцінювання якості побудованої мережі Байєса? Наведіть формули структурної різниці та перехресної ентропії.
2. В чому полягає різниця між евристичними та точними методами побудови мереж Байєса?
3. Який вигляд має функція опису мінімальної довжини (ОМД)? Наведіть означення, формулу, приклад.
4. Міра Кулбака-Леблера: дайте визначення, наведіть формулу для обчислення та поясніть її зміст.
5. Дайте означення структури математичної моделі.

**ПРАКТИЧНА РОБОТА 4**

**Мета:** Навчитися будувати експертні системи із використанням сучасних програм для побудови мереж Байєса на основі статистичних даних опитувань.

**Завдання.**

**Умова задачі.** В рамках дослідження було проведено анкетування студентів старших курсів університету щодо їхнього задоволення життям. Анкета містила наступні запитання:

1. Чи берете участь в святкових заходах? 1 – так, 0 – ні.
2. Ви достатньо розумний? 1 – так, 0 – ні.
3. Ви креативний? 1 – так, 0 – ні.
4. Ви встигаєте виконувати всі домашні завдання? 1 – так, 0 – ні.
5. Ви маєте власний ноутбук? 1 – так, 0 – ні.

6. Чи задоволені Ви навчанням в університеті? 1 – так, 0 – ні.

7. Ви вже працюєте паралельно із навчанням? 1 – так, 0 – ні.

8. Ви відчуваєте себе щасливою людиною? 1 – так, 0 – ні.

**Результати анкетування 5000 студентів можна знайти в файлі students.csv** (або завантажити з мережі Інтернет: [www.cs.cmu.edu/~ggordon/10601/hws/hw2/students.csv.zip](http://www.cs.cmu.edu/~ggordon/10601/hws/hw2/students.csv.zip)). Кожний рядок файлу **students.csv** відповідає одній анкеті опитування. Всього вісім запитань, відповіді на які розташовані у вигляді стовпчиків таблиці, поіменованих в наступній послідовності: (1) Party; (2) Smart; (3) Creative; (4) HW; (5) Mac; (6) Project; (7) Success; (8) Happy, відповідно. **Після обробки результатів** опитування із залученням психологів були **отримані наступні залежності:**

1. HW залежить тільки від змінних Party та Smart.

2. Mac залежить тільки від змінних Smart та Creative.

3. Project залежить тільки від Smart та Creative.

4. Success залежить тільки від HW та Project.

5. Happy залежить тільки від Party, Mac та Success.

**1. Маючи експертні рекомендації психолога побудуйте мережу Байєса** та наведіть її в звіті.

**2. Наведіть** в звіті формулу обчислення загальної сумісної ймовірності побудованої мережі Байєса.

**3. Використовуючи дані опитувань (файл students.csv), побудуйте в програмі Bayes Net** або будь-якій іншій програмі для роботи з мережами Байєса (наприклад, *Netica*), мережу Байєса та обчисліть наступні значення ймовірностей:

$P(\text{creative}=1) = ?$

$P(\text{smart}=1) = ?$

$P(\text{party}=1) = ?$

$P(\text{happy}=1) = ?$

$P(\text{happy}=1) = ?$  за умови, що party=1, smart=1, creative=0.

$P(\text{happy}=1) = ?$  за умови, що party=0, hw=1, project=1.

$P(\text{happy}=1) = ?$  за умови, що mac=1.

$P(\text{party}=1) = ?$  за умови, що smart=1, happy=1.

**4. Визначте** кількість незалежних параметрів, що необхідна для побудови кожної таблиці умовної ймовірності кожної вершини мережі Байєса. Результат слід представити у вигляді таблиці.

Вершина	Кількість параметрів
$P(\text{creative})$	
$P(\text{smart})$	
$P(\text{party})$	
$P(\text{project} \mid \dots)$	
$P(\text{mac} \mid \dots)$	
$P(\text{hw} \mid \dots)$	
$P(\text{success} \mid \dots)$	
$P(\text{happy} \mid \dots)$	

### Контрольні запитання

1. Поясніть своїми словами, що означає термін «ймовірнісний висновок».
2. Дайте означення ймовірнісного висновку у мережах Байєса. Наведіть приклади.
3. Наведіть методи обчислення якості побудови ймовірнісного висновку в мережах Байєса.
4. Які існують групи методів обчислення ймовірнісного висновку?
5. Які існують методи та алгоритми формування точного ймовірнісного висновку?
6. Наведіть приклади статистичних характеристик для оцінювання якості роботи будь-якого методу побудови ймовірнісного висновку.
7. Яким ще може бути тип висновку, крім ймовірнісного?

### ПРАКТИЧНА РОБОТА 5

**Мета:** Навчитися будувати системи підтримки прийняття рішень використовуючи сучасні програми для побудови мереж Байєса на основі логіко-ймовірнісної вхідної інформації

#### **Завдання.**

**Умова задачі.** Фінансово-кредитна установа намагається побудувати систему попередження шахрайства серед своїх клієнтів. В розпорядженні є наступна інформація:

- клієнти, що вирушають у подорож за кордон – більш ймовірні жертви зловмисників, що прагнуть вкрати гроші з карткового рахунку завдяки шахрайським транзакціям. Як наслідок близько 0,5% транзакцій є шахрайськими, якщо власник картки подорожує закордоном. В той самий час, за нормальних обставин, коли власник картки не подорожує за кордоном – лише 0,1% транзакцій шахрайські;

- в середньому 5% від усіх транзакцій, що здійснюються всіма клієнтами здійснюються під час подорожі власником картки;

- якщо транзакція шахрайська, то ймовірність здійснення покупки за кордоном збільшується, якщо тільки власник картки не подорожує за кордоном. Коли власник картки не подорожує за кордоном, то 10% транзакцій з придбання товарів за кордоном є шахрайськими, в той час як лише 1% транзакцій з покупки товарів за кордоном є легітимний;

- коли власник картки подорожує, то лише 90% транзакцій з придбання товарів за кордоном є легітимними;

- покупки, що здійснюються через мережу Інтернет, більш ймовірно шахрайські. Найчастіше це відповідає дійсності, коли власник картки не є власником комп'ютера. Сьогодні 60% населення має власний комп'ютер. 1% транзакцій здійснюється через Інтернет. 2% транзакцій, що здійснюються через Інтернет є шахрайськими;

- клієнти, що не мають свого власного комп'ютера, 0,1% транзакцій здійснюють через Інтернет, але при цьому відбувається 1,1% шахрайських транзакцій. На жаль, фінансова установа не має інформації чи має клієнт власний комп'ютер;



- щотижня 10% клієнтів, що мають власний комп'ютер здійснюють покупки через Інтернет. В той час, як тих, хто не має власного комп'ютера здійснює покупку через Інтернет 0,1%.

**1. Побудуйте** мережу Байеса для виявлення шахрайських транзакцій. Для початку необхідно побудувати на основі вищенаведеної інформації таблиці умовних ймовірностей. Мережа повинна містити саме шість вершин:

1. OC (owns computer) – клієнт має власний комп'ютер.
2. Fraud – транзакція шахрайська.
3. Trav – клієнт на даний момент часу подорожує за кордоном.
4. FP – поточна транзакція пов'язана з покупкою за кордоном.
5. IP – поточна транзакція є покупкою через мережу Інтернет.
6. CRP (a computer related purchase) – покупка здійснена із використанням комп'ютера протягом останньої неділі.

Дуги мережі Байеса повинні відображати ймовірнісні залежності між наведеними вершинами.

**2. Визначте** значення апіорної ймовірності того, що транзакція є шахрайською.

Визначте, **значення ймовірності** того, що транзакція шахрайська, за умови:

- транзакція здійснена за кордоном,
- через мережу Інтернет,
- за останній тиждень власник картки здійснював покупку через Інтернет з власного комп'ютера.

**3. Яким чином зміниться** значення ймовірності того, що транзакція шахрайська, після одержання вищенаведеної нової порції інформації, якщо система виявлення шахрайства повідомила про підозрілу транзакцію, а працівник відділу безпеки банку спробувавши зв'язатися з власником картки, виявив, що власник відсутній дома, разом з тим, родичі власника картки повідомили, що він у бізнес-подорожі за кордоном.

**4. Що має зробити** шахрай для того, щоб зменшити ризик виявлення шахрайської транзакції, у випадку, якщо виявився шахраєм працівник банку, який вкрав кредитну картку і знає, що система виявлення шахрайських транзакцій використовує мережу Байеса. Шахрай прагне здійснити крупну шахрайську покупку через мережу Інтернет. **Обґрунтуйте відповідь.**

### Контрольні запитання

1. Які методи та алгоритми формування апроксимаційного ймовірнісного висновку існують?
2. Що означає «довіра» стосовно мережі у цілому?
3. Поясніть термін «ініціалізація дерева».
4. У чому полягає алгоритм виключення змінних? Для яких структур він застосовується? Вкажіть переваги та недоліки.
5. У чому полягає суть алгоритму символічного ймовірнісного висновку? Для яких структур він використовується? Опишіть переваги та недоліки.

6. Чим можна пояснити неоднозначність ймовірнісних висновків, отриманих за допомогою різних методів?

7. В чому полягає алгоритм ймовірнісного висновку на основі диференційного підходу? Для яких структур він використовується? Опишіть його переваги та недоліки.

8. Поясніть алгоритм формування точного ймовірнісного висновку на основі навчальних даних. Наведіть приклад використання.

9. Що означає нормування значень ймовірностей?

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Чубукова И. А. Data Mining / И. А. Чубукова – М.: Бином ЛБЗ, 2008. – 384 с.
2. Wu X et al. Top 10 algorithms in data mining // Knowledge and Information Systems. – 2008. – Vol. 14., №1. – P. 1-37.
3. Терехов С. А. Вероятностное моделирование в байесовых сетях / С. А. Терехов // Лекция для школы-семинара “Современные проблемы нейроинформатики”, 29-31 января 2003 г. – М.: МИФИ, 2003 г. – 63 с.
4. Bayes T. (1763). An Essay towards solving a Problem in the Doctrine of Chances. By the late Rev. Mr. Bayes, F.R.S., communicated by Mr. Price, in a letter to John Canton, A.M. and F.R.S. // Philosophical Transactions of the Royal Society of London. – 1763. – 53. – P. 370–418.
5. Згуровський М. З. Системна методика побудови байесових мереж / М.З. Згуровський, П.І. Бідюк, О.М. Терентьев // “Наукові вісті” НТУУ “КПІ”. – 2007. – №4. – С. 47–61.
6. Бидюк П. И. Метод вероятностного вывода в байесовских сетях по обучающим данным / П.И. Бидюк, А.Н. Терентьев // Кибернетика и системный анализ. – 2007. – № 3. – С. 93-99.
7. Тулупьев А. Л. Байесовские сети: логико-вероятностный подход / А. Л. Тулупьев, С. И. Николенко, А. В. Сироткин – СПб.: Наука, 2006. – 607 с.
8. Pearl J. Causality: models, reasoning, and inference. – Cambridge University Press, 2000. – 400 p.
9. Gorodetsky V.I., Drozdgin V.V., Jusupov R.M. Application of attributed grammar and algorithmic sensitivity model for knowledge representation and estimation // Artificial Intelligence and Information, Control Systems of ROBOTSA. – North Holland: Elsevier Science, 1984. – P. 232-237.
10. Городецкий В.И. Байесовский вывод / В.И. Городецкий. – Л.: ЛИИАН, 1991. – 149 с.
11. Городецкий В.И. Алгебраические байесовские сети – новая парадигма экспертных систем / В.И. Городецкий // Юбилейный сборник трудов институтов Отделения информатики, вычислительной техники и автоматизации РАН. – М.: РАН, 1993. – Т. 2. – С. 120-141.
12. Тулупьев А. Л. Байесовские сети: вероятностная семантика и оптимизационные алгоритмы в логико-вероятностном выводе / А. Л. Тулупьев, Д. А. Никитин, С. И. Николенко / Материалы семинара “Информатика и компьютерные технологии”, Санкт-Петербургский институт информатики и автоматизации Российской Академии наук. – 2004. – 74 с.
13. Тулупьев А. Л. Лекция 5: Введение в байесовские сети – логико-вероятностная модель баз фрагментов знаний с неопределенностью / А. Л. Тулупьев, А. В. Сироткин // Курс лекций по предмету “Алгоритмы для Интернета”, Математико-Механический факультет, Санкт-Петербургский государственный университет информационных технологий. – 2006. – 184 с.
14. Терентьев А.Н. Эвристический метод построения байесовских сетей / А.Н. Терентьев, П.И. Бидюк / Интеллектуальные системы принятия решений и прикладные аспекты информационных технологий: сб. науч. трудов по материалам II междунар. конф., 15-18 мая 2006 г., Евпатория. – Херсон: ХМИ, 2006. – Т. 1. – С. 401–403.

15. Гасанова Л.Т. Застосування мережі Байєса в медицині / Л.Т. Гасанова, О.М. Терентьев, І.В. Мельник // Вісник Університету "Україна". – К.: Університет "Україна", 2008. – № 6. – С. 93-96.
16. Heckerman D. E., Horvitz E. J. and Nathwani B. N. Toward normative expert systems: Part I The PathFinder Project // Methods of information in medicine. – 1992. – Vol. 31, №2. – P. 90-105.
17. Бідюк П. І. Комп'ютерні системи підтримки прийняття рішень : навч. посіб. / П. І. Бідюк, О. П. Гожий, Л. О. Коршевич; Нац. техн. ун-т України "Київ. політехн. ін-т", Чорномор. держ. ун-т ім. Петра Могили. - Миколаїв. - К., 2012. - 379 с.
18. Барсегян, А. А. Анализ данных и процессов: учеб. пособие / А. А. Барсегян, М. С. Куприянов, И. И. Холод, М. Д. Тесс, С. И. Елизаров. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2009. — 512 с.: ил. + CD-ROM ISBN 978-5-9775-0368-6
19. Офіційний сайт щорічної міжнародної конференції IEEE-ICDM – Електронний ресурс: - Режим доступу <http://icdm2014.sfu.ca/home.html>
20. Data Mining – Електронний ресурс: - Режим доступу <http://bug.kpi.ua/stud/work/RGR/DATAMINING/whatisdm.html>
21. Офіційний сайт фірми Gartner – Електронний ресурс: - Режим доступу <http://www.gartner.com/technology/research/magic-quadrants>
22. Edelstein Herbert A. Introduction to Data Mining and Knowledge Discovery. Third Edition Paperback. – Potomac.: Two Crows Corporation, 1999. – 36 p.
23. Horvitz E., Ruokangas C. and Srinivas S. A decision-theoretic approach to the display of information for time-critical decisions: the Vista Project. [Text] / In Proceedings of annual workshop on Space Operations, Applications and Research (SOAR'92), NASA/Johnson Space Center, Houston, Texas, US, 4-6 August, 1992. – NY.: Elsevier science, 1992. – P. 407-417.
24. Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. – SF.: Morgan Kaufmann, September 1988. – 552 p.
25. Burnell L. and Horvitz E. Structure and chance: melding logic and probability for software debugging // Communications of the ACM (CACM). – 1995. – Vol. 38, № 3. – P. 31-57.
26. Кульчицький О. Основи філософії і філософічних наук / Кульчицький О. – Львів, 1995.
27. Wright S. Correlation and Causation / Journal of Agricultural Research, 20. – 1934. – P. 557-585.
28. Herbert S. Books by Herbert A. Simon [Електронний ресурс]. – Режим доступу: <http://www.amazon.com/Herbert-A.-Simon/e/B001HCWQ7A>
29. Terentyev A. N. Bayesian networks in credit scoring / A. N. Terentyev, L. T. Gasanova / The second international conference on control and optimization with industrial applications (COIA-1008), June 2-8, 2008, Baku, Azerbaijan. – Baku: Institute of applied mathematics BSU, 2008. – P. 171.
30. Zweig G.G. Speech recognition with dynamic Bayesian networks / Proceedings of the fifteenth conference on artificial intelligence, Madison (Wisconsin US). – 1998. – P. 173-180.
31. Murphy K. A brief introduction to graphical models and Bayesian networks / Technical report 2001-5-10, department of computer science,

32. Murphy K.P. Dynamic Bayesian networks: representation, inference and learning / A PhD dissertation, University of California, Berkeley. – 2002. – 225 p.
33. Buntine W. L. A Guide to the literature on learning probabilistic networks from data // IEEE Transactions on knowledge and data engineering. – Piscataway: IEEE Educational Activities Department, 1996. – Vol. 8, № 2. – P. 195-210.
34. Hautaniemi S.K., Petri T. Korpisaari and Jukka P.P. Saarinen. Target identification with dynamic hybrid Bayesian networks / Proceedings of the forth SPIE's international symposium on sensor fusion: architectures, algorithms and applications, Orlando (Florida US), April 2000. – P. 55 – 66.
35. Nodelman U., Christian R. Shelton, and Daphne Koller. Learning continuous time Bayesian networks / Proceedings of the nineteenth international conference on Uncertainty in Artificial Intelligence (UAI'03), Acapulco (Mexico), 7–10 August, 2003. – SF.: Morgan Kaufmann, 2003. – P. 451-458.
36. Elidan G. and Friedman N. Learning Hidden Variable Networks: The Information Bottleneck Approach [Text] / Journal of Machine Learning Research, 6. -2005. – P. 81-127.
37. Cheng J., Bell D.A. and Liu W. Learning belief networks from data: an information theory based approach / Proceedings of the sixth international conference on information and knowledge management (CIKM 1997), Las Vegas (Nevada), November 10-14. – 1997. – P.325-331.
38. Cheng J., Greiner R., Kelly J., Bell D.A. and Liu W. Learning Bayesian networks from data: an information-theory based approach // The artificial intelligence journal (AIJ). – 2002. – 137. – P. 43-90.
39. Russell S. J. Artificial Intelligence: A Modern Approach (2nd Edition) [Text] / Stuart J. Russel, Peter Norvig. – New Jersey.: Prentice Hall, 2002. – 1132 p.
40. Golmard J.-L., Mallet A. Learning probabilities in causal trees from incomplete databases [Text] / Revue d'Intelligence Artificielle, 5. – 1991. – P.93–106.
41. Spiegelhalter D.J., Cowell R.G. Learning in probabilistic expert systems [Text] / Bayesian Statistics, vol. 4. – Oxford, UK:Oxford University Press, 1992. – P. 447-465.
42. Spiegelhalter D., Dawid P., Lauritzen S. and Cowell R. Bayesian analysis in expert systems [Text] / Statistical Science, 8. – 1993. – P.219–282.
43. Olesen K.G., Lauritzen S.L., Jensen F.V. aHUGIN: A system for creating adaptive causal probabilistic networks [Text] / Uncertainty in Artificial Intelligence: Proceedings of the Eighth Conference. – CA: MorganKaufmann, 1992. – P. 223-229/
44. Dempster A.P., Laird N.M. and Rubin D. B. Maximum likelihood from incomplete data via the EM algorithm // Journal of the Royal Statistical Society. – 1977. – Vol. 39, №1. – P. 1-38.
45. Friedman N. Learning belief networks in the presence of missing values and hidden variables [Text] / Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997. – SF.: Morgan Kaufmann, 1997. – P. 125-133.
46. Gilks W. R., Richardson S., Spiegelhalter D. J. Markov chain Monte Carlo methods in practice [Text] / New York: Chapman Hall. – 1996.

47. Spiegelhalter D.J. and Lauritzen S.L. Sequential updating of conditional probabilities on directed graphical structures [Text] / *Networks*, 20. – 1990. – P.579–605.
48. Binder J., Koller D., Russell S. and Kanazawa K. Adaptive probabilistic networks with hidden variables [Text] / *Machine Learning*. – 1997. – Vol. 29. – P. 213–244.
49. Russell S., Binder J., Koller D. and Kanazawa K. Local learning in probabilistic networks with hidden variables [Text] / *Fourteenth International Joint Conference on Artificial Intelligence, Montreal, Canada, August 20-25 1995*. – 1995. – P.1146-1152.
50. Thiesson B. Accelerated quantification of Bayesian networks with incomplete data [Text] / *In Proceedings of First International Conference on Knowledge Discovery and Data Mining, Montreal, QU*. – Morgan Kaufmann, 1995. – P.306-311.
51. Lauritzen S. L. The EM algorithm for graphical association models with missing data [Text] / *Computational Statistics and Data Analysis*. – 1995. – Vol. 19. – P. 191-201
52. Friedman N. The Bayesian structural EM algorithm / *Fourteenth conference on Uncertainty in Artificial Intelligence (UAI'98), Madison, Wisconsin, USA, 24–26 July, 1998*. – SF.: Morgan Kaufmann, 1998. – P. 129-138
53. Ramoni M. and Sebastiani P. Learning Bayesian Networks from Incomplete Databases [Text] / *Thirteenth Conference on Uncertainty in Artificial Intelligence, Providence, Rhode Island, USA, August 1-3, 1997*. - San Mateo (CA).: Morgan Kaufmann, 1997. – P.401-408.
54. Ramoni, M.F., Sebastiani, P. Parameter estimation in bayesian networks from incomplete databases [Text] / *Intelligent Data Analysis*, 2. – 1998. – P.139-160.
55. Beinlich I.A., Suermondt H.J., Chavez R.M., and Cooper G.F. The ALARM Monitoring System: A Case Study With Two Probabilistic Inference Techniques for Belief Networks [Text] / *Second European Conference on Artificial Intelligence in Medicine*. – 1989. – P. 247-256.
56. Lauritzen S.L. and Spiegelhalter D.J. Local computations with probabilities on graphical structures and their application to expert systems // *Journal Royal Statistics Society, series B – statistical methodology*. – 1988. – Vol. 50, №2. – 157-194 p.
57. Guo H. Real time Bayesian networks inference / *Report on research activities, Kansas State University, department of computing and information sciences, March November 2001*. – 42 p.
58. Guo H. and Hsu W. A survey of algorithms for real-time Bayesian network inference / *In the joint AAAI-02/KDD-02/UAI-02 workshop on real-time decision support and diagnosis systems, Edmonton, Alberta, Canada, 1–4 August, 2002*. – SF.: Morgan Kaufmann, 2002. – P. 1-12.
59. Pearl J. Fusion, propagation and structuring in belief networks // *Artificial Intelligence*, imprint: Elsevier, September 1986. – Vol. 29, №3. – P. 241–288.
60. Hautaniemi S.K. Target identification with Bayesian networks / *Master of science thesis, Tampere University of Technology, Finland, 2002*. – 87 p.
61. Guo H. Probabilistic inference in Bayes networks / *Report on research activities, Kansas State University, department of computing and information sciences, November 2000*. – 27 p.

62. Kozlov A.V. and Singh J.P. A parallel Lauritzen-Spiegelhalter algorithm for probabilistic inference / Proceedings of the Supercomputing'94 conference, Washington, DC, November 1994. – P. 320-329.
63. Jensen F.V., Olesen K. G. and Andersen S. K. An algebra of Bayesian belief universes for knowledge-based systems // Networks. – Wilmington: Wiley Periodicals Inc, 1990 – Vol. 20, №5. – P. 637-659.
64. Lauritzen S.L. and Jensen F.V. Local computation with valuations from a commutative semigroup // Annals of mathematics and artificial intelligence, – NY.: Springer, 1997. – Vol. 21, № 1. – P. 51-69.
65. Lepar V. and Shenoy P.P. A comparison of architectures for exact computation of marginals / Technical reports №274, school of business, the University of Kansas, February 1997. – 58 p.
66. Shenoy P.P. Binary join trees for computing marginals in the Shenoy-Shafer architecture // International Journal of approximate reasoning, publisher: Elsevier, August 1997. – Vol. 17, №2. – P. 239-263.
67. Lepar V. and Shenoy P.P. A comparison of Lauritzen-Spiegelhalter, Hugin, and Shenoy-Shafer architectures for computing marginals of probability distributions / Fourteenth conference on Uncertainty in Artificial Intelligence (UAI'98), Madison, Wisconsin, USA, 24–26 July, 1998. – SF.: Morgan Kaufmann, 1998. – P. 328-337.
68. Madsen A. L. and Jensen F. V. Lazy propagation in junction trees / Fourteenth conference on Uncertainty in Artificial Intelligence (UAI'98), Madison, Wisconsin, USA, 24–26 July, 1998. – SF.: Morgan Kaufmann, 1998. – P. 362-369.
69. Liu R. F. and Soetjipto R. Learning on Bayesian networks / Report for class project in course MIT 6.825: techniques in artificial intelligence, the MIT computer science and artificial intelligence laboratory, Massachusetts Institute of Technology, Cambridge, December 2004. – 39 p.
70. Bylander T. A variable elimination algorithm for belief networks / Course of lectures on “Artificial intelligence”, University of Texas at San Antonio, department of computer science. – 2007. – 4 p.
71. Castillo E., Gutierrez J.M. and Hadi A.S. Propagation in Bayesian networks (symbolic propagation) / Technical report 98-06-07, artificial intelligence research group from Cantabria University (Spain) and Cornell University (USA). – 1998. – 4 p.
72. Castillo E., Gutierrez J.M. and Hadi A.S. A new method for efficient symbolic propagation in discrete Bayesian networks // Networks. – Wilmington: Wiley Periodicals Inc, 1996. – Vol. 28. – P. 31-43.
73. Castillo E., Gutierrez J.M. and Hadi A.S. Goal oriented symbolic propagation in Bayesian networks / Thirteenth National Conference on Artificial Intelligence (AAAI-96), Portland (USA). – 1996. – Vol. 2. – P. 1263-1268.
74. Castillo E., Gutierrez J.M., Hadi A.S. and Solares C. Symbolic propagation and sensitivity analysis in Gaussian Bayesian networks with application to damage assessment // Journal of Artificial intelligence in engineering, publisher: Elsevier science. – 1997. – Vol. 11. – P. 173-181.
75. Shachter R. D. Evidence absorption and propagation through evidence reversals / Fifth workshop on Uncertainty in Artificial Intelligence (UAI'89), University of Windsor, Ontario, Canada, 8–10 August, 1989. – Amsterdam: North-Holland, 1990. – P. 303-310.

76. Aji S.M. and McEliece R.J. The generalized distributive law // IEEE transaction on information theory. – Piscataway: IEEE Educational Activities Department, 2000. – Vol. 46, № 2. – P. 325-343.
77. Gillies D.A. Lecture 11: Exact Inference / Course of lectures on “Intelligent data analysis and probabilistic inference”. Imperial college of London, department of computing. – 2006. – 4 p.
78. Gillies D.A. Lecture 12: Propagation of probabilities in join trees / Course of lectures on “Intelligent data analysis and probabilistic inference”. Imperial college of London, department of computing. – 2006. – 4 p.
79. Namasivayam V.K., Pathak A. and Prasanna V.K. Scalable parallel implementation of Bayesian network to junction tree conversion for exact inference / The eighteenth International symposium on computer architecture and high performance computing (SBAC-PAD'06), Brazil, October 17-20. – 2006. – P. 167-176.
80. Kevorkian A.K. Decomposition of large sparse symmetric systems for parallel computation. Part 1: theoretical foundations / Technical report №1572. Naval command, control and ocean surveillance center RDT&E division, San Diego, California, March 1993. – 56 p.
81. Tarjan R. E. and Yannakakis M. Simple linear time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs // SIAM Journal on computing (SICOMP), publisher: Society for industrial and applied mathematics (SIAM), August 1984. – Vol. 13, №3. – P. 566-579.
82. Bodlaender H.L. and Arie M.C. On the maximum cardinality search lower bound for tree width // Discrete applied mathematics, publisher: Elsevier, June 2007. – Vol. 155, № 11. – p. 1348-1372.
83. Berry A., Blair J. and Heggernes P. Maximum cardinality search for computing minimal triangulations of graphs // Algorithmica, publisher: Springer, May 2004. – Vol 39, №4. – P. 287-298.
84. Oellrich M. Algorithms for the construction of reliable network platforms in telecommunication networks / Master of Science thesis. The Technical University of Berlin, Germany, May 1997. – 244 p.
85. Dharwadker A. The clique algorithm / Technical report № H-501, Institute of Technology, Palam Vihar, state Haryana, India. – 2006. – 41 p.
86. Tung L. A clique tree algorithm exploiting context specific independence / Master of science thesis, the university of British Columbia, Canada, August 2002. – 92 p.
87. Huang C. and Darwiche A. Inference in belief networks: a procedural guide // International journal of approximate reasoning. – 1996. – Vol. 15, №3. – P. 225-263.
88. Bidyuk B.P. Exploiting graph cutsets for sampling-based approximations in Bayesian networks / A PhD dissertation, University of California, Irvine, July 2006. – 254 p.
89. Diez F. J. Local conditioning in Bayesian Networks // Artificial Intelligence. – 1996. – Vol. 87, №1. – P. 1-20.
90. Kim J.H. and Pearl J. A computational model for causal and diagnostic reasoning in inference engines / Proceedings of the eighth International Joint Conference on



- Artificial Intelligence (IJCAI'83), Karlsruhe, West Germany, 8-12 August, 1983. – Los Altos: William Kaufmann, 1983. – Vol. 1 – P. 190-193.
91. Shachter R. D., Anderson S. K. and Szolovits P. Global conditioning for probabilistic inference in belief networks / Proceedings of the tenth annual conference on Uncertainty in Artificial Intelligence (UAI'94), Seattle, Washington, USA, 29–31 July, 1994. – SF.: Morgan Kaufmann, 1994. – P.514-522.
  92. Darwiche A. Conditioning methods for exact and approximate inference in causal networks / Proceedings of the eleventh annual conference on Uncertainty in Artificial Intelligence (UAI'95), Montreal, Quebec, Canada, 18–20 August, 1995. – SF.: Morgan Kaufmann, 1995. – P. 99-107.
  93. Darwiche A. Recursive conditioning // Artificial Intelligence. – 2001. – Vol 125, №1. – P. 5-41.
  94. Robinson R.W. Counting unlabeled acyclic digraphs / Proceeding of fifth Australian on combinatorial mathematics. Melbourne, Australia. – 1976. – P. 28-43.
  95. Dechter R. Bucket elimination: a unifying framework for reasoning // ACM Press, December 1996. –Vol. 28, № 61. – P. 1-51.
  96. Liu R. F. and Soetjipto R. Analysis of three Bayesian network inference algorithms: variable elimination, likelihood weighting and Gibbs sampling / Report for class project in course MIT 6.825: techniques in artificial intelligence, the MIT computer science and artificial intelligence laboratory, Massachusetts Institute of Technology, Cambridge, November 2004. – 35 p.
  97. Kuminov D. Lecture: Variable elimination algorithm / Course of lectures on “Introduction to Artificial Intelligence”, Faculty of Industrial Engineering & Management, Technion - Israel Institute of Technology, September 2006. – 13 p.
  98. Stracuzzi D. J. and Utgoff P. E. Randomized Variable Elimination // Journal of Machine Learning Research. – 2004. – 5. – P. 1331–1362.
  99. Cozman F. G. Generalizing variable elimination in Bayesian networks / Proceedings of the seventh Ibero-American conference on Artificial Intelligence, fifteenth Brazilian Symposium on Artificial Intelligence (IBERAMIA-SBIA-2000), workshop on probabilistic reasoning in Artificial Intelligence, Atibaia, Brazil. – 2000. – P. 27-32.
  100. Li Z. and D'Ambrosio B. Efficient Inference in Bayes Nets as a combinatorial optimization problem // International Journal of Approximate Reasoning, publisher: Elsevier science. –1994. – Vol 11, № 1. – P. 55-81.
  101. D'Ambrosio B. Local expression languages for probabilistic dependence // International journal of approximate reasoning, July 1995. – Vol 13, № 1. – P. 61-81.
  102. Darwiche A.. A differential approach to inference in Bayesian networks // Journal of the association for computing machinery (JACM). – 2003 – Vol. 50, №3. – P. 280-305.
  103. Gaag L.C. On evidence absorption for belief networks / Technical report CS-RUU-93-35, department of computer science, Utrecht University, Netherlands. – 1993. – 17 p.
  104. Jaakkola T.S. and Jordan M.I. Variational Probabilistic Inference and the QMR-DT network // Journal of artificial intelligence research (JAIR). – Menlo Park: AAI Press, 1999. – 10. – P. 291-322.

105. Carlin B. P., Gelman A. and Neal R. M. Chain Monte Carlo in practice: a roundtable discussion // *The American statistician*. – Alexandria: American Statistical Association, 1998. – Vol. 52, № 2. – P. 93-100.
106. Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H., and Teller E. Equations of state calculations by fast computing machines // *Journal of Chemical Physics*. – 1953. – Vol. 21, № 6. P. 1087-1092.
107. Geman S. and Geman D. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images // *IEEE transactions on pattern analysis and machine intelligence*. – Washington: IEEE Computer Society, 1984. – Vol. 6, №6. – P. 721-741.
108. Бідюк П.І., Меняйленко О.С., Половцев О.В. Методи прогнозування. – Луганськ: Альма-Матер, 2008. – 607 с.
109. Cornell University Library [Електронний ресурс]. – Режим доступу : <http://arxiv.org/abs/1504.01896>
110. Henrion M. Propagating uncertainty in Bayesian networks by probabilistic logic sampling / *Proceedings of the second international conference on Uncertainty in Artificial Intelligence (UAI'86)*, Philadelphia, Pennsylvania, USA, 8–10 August, 1986. – NY.: Elsevier science, 1988. – P. 149-163.
111. Henrion M. Search-based methods to bound diagnostic probabilities in very large belief nets / *Proceedings of the seventh international conference on Uncertainty in Artificial Intelligence (UAI'91)*, Los Angeles, California, USA, 13–15 July, 1991. – SF.: Morgan Kaufmann, 1991. – P. 142-150.
112. Fung R. and Chang K. C. Weighting and integrating evidence for stochastic simulation in Bayesian networks / *Proceedings of the fifth international conference on Uncertainty in Artificial Intelligence (UAI'89)*, University of Windsor, Ontario, Canada, 8–10 August, 1989. – Amsterdam: North-Holland, 1990. – P. 209-219.
113. Shenoy P. and Shenoy P. Bayesian Network models of portfolio risk and return / *Proceedings of the sixth annual international conference Computational Finance 1999*, New York University's Stern School of Business. –Cambridge: MIT Press, 2000. – P. 87-106.
114. Fung R. and Favero B. D. Backward simulation in Bayesian networks / *Proceedings of the tenth annual conference on Uncertainty in Artificial Intelligence (UAI'94)*, Seattle, Washington, USA, 29–31 July, 1994. – SF.: Morgan Kaufmann, 1994. – P.227-234.
115. Gilks W. R. and Wild P. Adaptive rejection sampling for Gibbs sampling // *Applied Statistics*. – London: Routledge, 1992. – 41, № 2. – P. 337-348.
116. Gilks W. R. and Wild P. Algorithm AS 287: Adaptive rejection sampling from log-concave density // *Applied Statistics*. – London: Routledge, 1993. – 42, № 4. – P. 701-709.
117. Kohnen C. N. Adaptive Rejection Sampling / *Technical report № STA-378*, Duke University, Durham, England. – 2002. – 34 p.
118. Shachter R. D. and Peot M. A. Simulation approaches to general probabilistic inference on belief networks / *Proceedings of the sixth international conference on Uncertainty in Artificial Intelligence (UAI'90)*, Cambridge, Massachusetts, USA, 27–29 July, – NY.: Elsevier science, 1991. – P 221-234.

119. Cheng J. and Druzdzel M. AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks // *Journal of Artificial Intelligence Research (JAIR)*. – Menlo Park: AAAI Press, 2000. – 13. – P. 155-188.
120. Cheng J. and Druzdzel M. J. Confidence inference in Bayesian networks / *Proceedings of the seventeenth international conference on Uncertainty in Artificial Intelligence (UAI'01)*, Seattle, Washington, USA, 2–5 August, 2001. – SF.: Morgan Kaufmann, 2001. – P. 75-82.
121. Jensen F. and Andersen S. Approximations in Bayesian belief universes for knowledge-based systems / *Proceedings of the sixth international conference on Uncertainty in Artificial Intelligence (UAI'90)*, Cambridge, Massachusetts, USA, 27–29 July, 1990. – NY.: Elsevier science, 1991. – P. 162-169.
122. Kjaerulff U. Reduction of computation complexity in Bayesian networks through removal of weak dependencies / *Proceedings of the tenth annual conference on Uncertainty in Artificial Intelligence (UAI'94)*, Seattle, Washington, USA, 29–31 July, 1994. – SF.: Morgan Kaufmann, 1994. – P. 374-382.
123. Engelen R. A. Approximating Bayesian belief networks by arc removal // *IEEE transactions on pattern analysis and machine intelligence*. – Washington: IEEE Computer Society, 1997. – Vol. 19, №8. – P. 916-920.
124. Friedman N. Inference III: approximate inference / *Course of lectures on “Automated planning and decision making”*, department of computer science Ben-Gurion University, Israel. – 2007. – 45 p.
125. Horvitz E.J., Suermondt H.J. and Cooper G.F. Bounded conditioning: flexible inference for decisions under scarce resources / *Proceedings of the fifth annual conference on Uncertainty in Artificial Intelligence (UAI'89)*, University of Windsor, Ontario, Canada, 8–10 August, 1989. – Amsterdam: North-Holland, 1990. – P. 182-193.
126. Draper D. and Hanks S. Localized partial evaluation of belief networks / *Proceedings of the tenth annual conference on Uncertainty in Artificial Intelligence (UAI'94)*, Seattle, Washington, USA, 29–31 July, 1994. – SF.: Morgan Kaufmann, 1994. – P. 170-177.
127. D'Ambrosio B. Incremental probabilistic inference / *Proceedings of the ninth annual conference on Uncertainty in Artificial Intelligence (UAI'93)*, Washington, Columbia, USA, 9–11 July, 1993. – SF.: Morgan Kaufmann, 1993. – P. 301-308.
128. Poole D. and Zhang N. L. Exploiting contextual independence in probabilistic inference // *Journal of Artificial Intelligence Research (JAIR)*. – Menlo Park: AAAI Press, 2003. – 18. – P. 263-313.
129. Dechter R. Mini-buckets: a general scheme for generating approximations in automated reasoning in probabilistic inference / *Proceedings of the fifteenth International Joint Conference on Artificial Intelligence (IJCAI'97)*, Nagoya, Japan, August 23-29, 1997. – NY.: Springer, 1999. – P. 1297-1302.
130. Sarkar S. Using tree-decomposable structures to approximate belief networks / *Proceedings of the ninth annual conference on Uncertainty in Artificial Intelligence (UAI'93)*, Washington, Columbia, USA, 9–11 July, 1993. – SF.: Morgan Kaufmann, 1993. – P. 376-382.
131. Wellman M. P. and Liu C. L. State-space abstraction for anytime evaluation of probabilistic networks / *Proceedings of the tenth annual conference on*

- Uncertainty in Artificial Intelligence (UAI'94), Seattle, Washington, USA, 29–31 July, 1994. – SF.: Morgan Kaufmann, 1994. – P.567-574.
132. Ghahramani Z. G. and Beal M. J. Propagation algorithms for variational Bayesian learning / Proceedings of the thirteenth conference on Neural Information Processing Systems (NIPS'2001), Vancouver, British Columbia, Canada, 3–8 December, 2001. – Cambridge: MIT Press, 1999. – P. 507-513.
  133. Jordan M. I., Ghahramani Z. G., Jaakkola T.S. and Saul L. K. An introduction to variational methods for graphical models / Machine Learning. – Cambridge: MIT Press, 1999. – Vol. 37, №2. – P. 183-233.
  134. Lin Y. and Druzdzel M. Stochastic sampling and search in belief updating algorithms for very large Bayesian networks / In working notes of the AAAI spring symposium on search techniques for problem solving under uncertainty and incomplete information, Stanford University, Stanford, California, USA, March 1999. – P. 77-82.
  135. Cooper G. F. A computer-based medical diagnostic aid that integrates causal and probabilistic knowledge / A PhD dissertation, Stanford University of California, department of computer science, Stanford. – 1984. – 256 p.
  136. D. Poole. The use of conflicts in searching Bayesian networks / Technical report TR-93-14, department of computer science, University of British Columbia, Canada. – 1993. – 9 p.
  137. Santos E. J., Shimony S. E., Solomon E., and Williams E. On a distributed anytime architecture for probabilistic reasoning / Technical report TR-95-02, department of electrical and computer engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, USA. – 1995. – 18 p.
  138. Santos E. J. and Shimony S. E. Deterministic approximation of marginal probabilities in bayes nets // IEEE Transactions on Systems, Man, and Cybernetics. – 1998. – Vol. 28, № 4. – P. 377-393.
  139. Suzuki J. Learning Bayesian belief networks based on the MDL principle: an efficient algorithm using the branch and bound technique. // IEICE Transaction on information and systems, February 1999. – P. 356-367.
  140. Suzuki J. Learning Bayesian belief networks based on the minimum description length principle: basic properties. // IEICE Transaction. on fundamentals, September 1999. – Vol. E82-A, № 9. – 9 p.
  141. Terent'yev A. N., Bidyuk P. I., Korshevnyuk L. A. Bayesian network as instrument of intelligent data analysis [Text] / Journal of Automation and Information Sciences. – New York: Begell House Inc., 2007. – 39. – P. 28-38.
  142. Lokeswarappa K. G. Junction trees: motivation [Text] / Seminar CSE 714 on advanced topics in machine learning, March 2005. – 57 p.
  143. Park J., Darwiche A. Morphing the Hugin and Shenoy-Shafer Architectures. [Text] / Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 7th European Conference, ECSQARU 2003, Aalborg, Denmark, July 2-5, 2003. – P. 149-160.
  144. Spirtes, P., Glymour, C. and Scheines, R., Causation, prediction and search // Adaptive computation and machine learning, MIT press, January 2001. – 565 p.
  145. Jouffe L. and Munteanu P. New search strategies for learning Bayesian networks / Proceedings of tenth international symposium on applied stochastic

- models and data analysis (ASMDA 2001). – Compiègne (France). 12–15 June 2001. – Vol. 2 – P. 591–596.
146. Chow C.K., Liu C.N. Approximating discrete probability distributions with dependence trees // *IEEE Transactions on information theory*, May 1968. – Vol. IT-14, №3. – P. 462 – 467.
  147. Rebane G., Pearl J. The recovery of causal poly-trees from statistical data. // *International journal of approximate reasoning*, July 1988. – Vol. 2, № 3. – P. 175-182
  148. Herskovits E. and Cooper, G. Kutato: an entropy-driven system for construction of probabilistic expert systems from databases / *Proceedings of the sixth international conference on Uncertainty in Artificial Intelligence (UAI'90)*, Cambridge, Massachusetts, USA, 27–29 July, – NY.: Elsevier science, 1991. – P 54-62.
  149. Cooper G., Herskovits E. A Bayesian method for the induction of probabilistic networks from data. // *Machine Learning*, 1992. – 9. – P. 309-347.
  150. Heckerman D., Geiger D. and Chickering D. Learning Bayesian networks: the combination of knowledge and statistical data / *Technical report MSR-TR-94-09*, Microsoft Research, March 1994. – 53 p.
  151. Wong, S. and Xiang, Y. Construction of a Markov network from data for probabilistic inference / *Third International workshop on rough sets and soft computing (RSSC'94)*, San Jose (California). – 1994. – P. 562-569.
  152. Chu T. and Xiang Y. Exploring parallelism in learning belief networks / *Proceedings of the thirteenth international conference on Uncertainty in Artificial Intelligence (UAI'97)*, Providence, Rhode Island, USA, 1–3 August, 1997. – SF.: Morgan Kaufmann, 1997. – P. 90-98.
  153. Lam W. and Bacchus F. Learning Bayesian belief networks: an approach based on the MDL principle. // *Computational Intelligence*, July 1994. – Vol. 10, № 4. – P. 269-293.
  154. Acid S. and Campos L. Benedict: an algorithm for learning probabilistic belief networks / *Proceedings of the sixth international conference IPMU'96*, Granada, Spain. – 1996. – P. 979–984.
  155. Singh M. and Valtorta M. Construction of Bayesian network structures from data: a brief survey and an efficient algorithm. // *International journal of approximate reasoning*, 1995. –12. – P. 111-131.
  156. Friedman N. and Goldszmidt M. Learning Bayesian networks with local structure. / *Proceedings of the twelfth international conference on Uncertainty in Artificial Intelligence (UAI'96)*, Portland, Oregon, USA, 1–4 August, 1996. – SF.: Morgan Kaufmann, 1996. – P. 252-262.
  157. Wallace C., Korb K. and Dai H. Causal discovery via MML / *Proceedings of the thirteenth international conference on machine learning (ICML'96)*, Bari, Italy. – SF.: Morgan Kaufmann, 1996. – P. 516-524.
  158. Бідюк П.І., Кузнецова Н.В., Терентьев О.М. Система підтримки прийняття рішень для аналізу фінансових даних // *Наукові вісті НТУУ “КПІ”*. – №1. –К.: НТУУ “КПІ” ВПІ ВПК “Політехніка”, 2011. – С. 48-61.
  159. Wermuth N. and Lauritzen S. Graphical and recursive models for contingency tables // *Biometrika*, December 1983. – Vol. 70, №3. – P. 537-552.

160. Srinivas S. Russell S. and Agogino A. Automated construction of sparse Bayesian networks from unstructured probabilistic models and domain information / Proceedings of the fifth annual conference on Uncertainty in Artificial Intelligence (UAI'90), Cambridge, Massachusetts, USA, 27–29 July, 1990. – NY.: Elsevier science, 1991. – P. 295 – 308.
161. Fung R.M., Crawford S.L. Constructor: a system for the induction of probabilistic models / Proceedings of the seventh national conference on artificial intelligence (AAAI-90). – 1990. – P. 762-769.
162. Madigan, D. and Raftery A. Model selection and accounting for model uncertainty in graphical models using Occam's window // Journal of the American statistical association (JASA). – 1994. – 89. P. 1535-1546.
163. Spirtes P., Glymour C. and Scheines R. An algorithm for fast recovery of sparse causal graphs // Social science computer review (SSCORE). – 1991. – 9. – P. 62-72.
164. Buntine W. Operations for learning with graphical models // Journal of artificial intelligence research (JAIR). – Menlo Park: AAAI Press, 1994. – 2. – P. 159-225.
165. Verma T. and Pearl J. Equivalence and synthesis of causal models / Proceedings of the sixth international conference on Uncertainty in Artificial Intelligence (UAI'90), Cambridge, Massachusetts, USA, 27–29 July, 1990. – NY.: Elsevier science, 1991. – P. 255-270.
166. Spirtes P., Richardson T. and Meek C. Heuristic greedy search algorithms for latent variable models / Proceedings of artificial intelligence and statistics (AI & Statistics 1997), Fort Lauderdale (Florida). – 1997. – P. 481-488.
167. Sebastiani P. and Ramoni M. Bayesian inference with missing data using bound and collapse // Journal of Computational and Graphical Statistics, December 2000. – Vol. 9, № 4. – P. 779-800.
168. Zhang Z, Kwok. J and Yeung D. Surrogate maximization (minimization) algorithms for AdaBoost and the logistic regression model / Proceedings of the twenty-first international conference on machine learning (ICML 2004). – 2004. – 117 p.
169. Leray P., Francois O. BNT structure learn package: documentation and experiments / Technical report, laboratory PSI-INSA Rouen-FRE CNRS 2645, November 2004. – 27 p.
170. Park J. and Darwiche A. Complexity results and approximation strategies for MAP explanations. // Journal of artificial intelligence research (JAIR). – Menlo Park: AAAI Press, 2004. – 21. – P. 101-133.
171. Agresti A. Categorical data Analysis. Publisher: John Wiley & Sons Canada Ltd. – 1990. – 576 p.
172. Васнев С.А. Статистика: учебное пособие / С.А. Васнев –Москва: МГУП, 2001. – 170 С.
173. Терентьев А. Н. Эвристический метод построения байесовских сетей / А.Н. Терентьев, П.И. Бидюк // Математические машины и системы. – К.: ТОВ “РВПК ЕКСЛИБРИС”, 2006. – № 3. – С. 12-23.
174. Шумский С.А. Байесова регуляризация обучения / С.А. Шумский // Лекции по нейроинформатике, часть 2. – М.: МИФИ, 2002. – 172 с.

175. Grunwald P. A Tutorial Introduction to the Minimum Description Length Principle. // *Advances in minimum description length: theory and applications*. MIT Press, Cambridge, MA, USA. – 2005. – 80 p.
176. Zheng Y. and Kwoh C.K. Improved MDL Score for Learning of Bayesian Networks / *Proceedings of the international conference on artificial intelligence in science and technology (AISAT 2004)*. – 2004. – P. 98-103.
177. Бидюк П. И. Построение и методы обучения байесовских сетей [Текст] / П. И. Бидюк, А. Н. Терентьев. -Таврический вестник информатики и математики. – Симферополь: КНЦ НАНУ, 2004. – № 2. – С. 139–153.
178. Луценко Е. В. Автоматизированный системно-когнитивный анализ в управлении активными объектами (системная теория информации и ее применение в исследовании экономических, социально-психологических, технологических и организационно-технических систем): Монография (научное издание). – Краснодар: КубГАУ. 2002. – 605 с.
179. Д. А. Поспелов Новые информационные технологии - это те ключи, которые откроют нам путь в новое общество // *Новости искусственного интеллекта*. - 1994. - 2. - С. 57-76.
180. Сайт Є. І. Соколова [Електронний ресурс]. – Режим доступу : <http://acs.nlu.edu.ua/ttablestd/index/>
181. Вагин, В. Н. Интеллектуальная система поддержки принятия решений "Аэропорт" [Текст] : научное издание / В.Н. Вагин, Е.Ю. Головина // *Авиакосмическое приборостроение : Ежемес. науч.-техн. и производственный журн.* - 2004. - №10. - С. 47-56.
182. Heckerman D. E., Horvitz E. J. and Nathwani B. N. Toward normative expert systems: Part II Probability-base representation for efficient knowledge acquisition and inference // *Methods of information in medicine*. – 1992. – Vol. 31, №2. – P. 106-116.
183. Kim J. H. and Pearl J. CONVINCENCE: A conversational inference consolidation engine. [Text] / *IEEE Transactions on systems, Man and Cybernetics*, 17(2): 120-132, 1983.
184. Shahar Y. Lecture 6: Computing with Influence Diagrams and the PathFinder Project / *Course of lectures on "Judgment and Decision Making in Information Systems"*. Ben-Gurion University of the Negev, department of Information System Engineering. – 2008. – 26 p.
185. Cooper G. F. NESTOR: a computer-based medical diagnostic aid that integrates causal and probabilistic knowledge. [Text] / *PhD thesis, Stanford University, Computer Science department*, 1984.
186. Andreassen S., Woldbye M., Falck B., and Andersen S.K. MUNIN – a causal probabilistic network for interpretation of electromyographic findings. [Text] / *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Los Altos, CA, Morgan Kaufmann Publishers, p. 366-373, 1987.
187. Friedman N., Ninio M., Pupko T. A structural EM algorithm for phylogenetic inference. [Text] / *Journal of Computational Biology*. - 2002. – Vol. 9. – P.331-353.
188. Shwe M., Middleton B., Heckerman D., Henrion M., Horvitz E., Lehmann H., and Cooper G. Probabilistic diagnosis using a reformulation of the

- INTERNIST- 1/QMR knowledge base I. The probabilistic model and inference algorithms. [Text] / *Methods of Information in Medicine*, 30(4): 241-255, 1991
189. Diez F. J., Mira J., Iturralde E., Zubillaga S. DIAVAL, a Bayesian expert system for echocardiography. [Text] / *Artificial Intelligence in Medicine*, 10: 59-73, 1997.
190. Mani S., McDermott S., Valtorta M. MENTOR: a Bayesian Model for prediction of mental retardation in newborns. [Text] / *Research in Developmental Disabilities*. – 1997. – Vol. 18, №5. – P.303-318.
191. Onisko A., Druzdzel M. J., and Wasyluk H. A probabilistic causal model for diagnosis of liver disorders. [Text] / In *Proceedings of the Seventh Symposium on Intelligent Information Systems (IIS-98)*, p. 379–387, Malbork, Poland, 1998.
192. Shahar Y. Lecture 6: Computing with Influence Diagrams and the PathFinder Project / Course of lectures on “Judgment and Decision Making in Information Systems”. Ben-Gurion University of the Negev, department of Information System Engineering. – 2008. – 26 p.
193. Sierra B, Larranaga P. Predicting survival in malignant skin melanoma using Bayesian networks automatically induced by genetic algorithms. An empirical comparison between different approaches. [Text] / *Artificial Intelligence in Medicine*, 14: 215-230, 1998.
194. Galan S. F., Aguado F., Diez F. J., Mira J. NasoNet, Joining Bayesian Networks and Time to Model Nasopharyngeal Cancer Spread. [Text] / *AIME 2001*, Springer-Verlag, Heidelberg, p. 207-216, 2001.
195. Mani S., McDermott S., Valtorta M. MENTOR: a Bayesian Model for prediction of mental retardation in newborns. [Text] / *Research in Developmental Disabilities*. – 1997. – Vol. 18, №5. – P.303-318.
196. Friedman N., Linia, M., Nachman I. Using Bayesian networks to analyze gene expression data. [Text] / *Journal of Computational Biology*. – 2000. – Vol. 7. – P.601-620.
197. Friedman N., Ninio M., Pupko T. A structural EM algorithm for phylogenetic inference. [Text] / *Journal of Computational Biology*. - 2002. – Vol. 9. – P.331-353.
198. Интеллектуальные информационные системы [Электронный ресурс]. – Режим доступа : [www.data-digest.com](http://www.data-digest.com)
199. Faculty of Computing, Engineering and Sciences [Электронный ресурс]. – Режим доступа : [www.soc.staffs.ac.uk ~cmtaa/bbnag.htm](http://www.soc.staffs.ac.uk/~cmtaa/bbnag.htm)).
200. Das B. Representing uncertainties using Bayesian Networks / Technical report DSTO-TR-0918, Defence Science and Technology Organization, Electronics and Surveillance Research Laboratory, Australian Department of Defense. – 1999. – 66 p.
201. Falzon L. Using Bayesian network analysis to support centre of gravity analysis in military planning // *European Journal of Operational Research*. – 2006. – Vol. 170, № 2. – P. 629-643.
202. Pourret O., Naim P., and Marcot B. Bayesian Networks: A Practical Guide to Applications. [Text] / Chichester, UK: Wiley. – 2008. – 448 p.
203. Horvitz E., Ruokangas C. and Srinivas S. A decision-theoretic approach to the display of information for time-critical decisions: the Vista Project / In *Proceedings of annual workshop on Space Operations, Applications and Research*



- (SOAR'92), NASA/Johnson Space Center, Houston, Texas, US, 4-6 August, 1992. – NY.: Elsevier science, 1992. – P. 407-417.
204. Horvitz E. and Barry M. Display of information for time-critical decision making / Proceedings of the eleventh annual conference on Uncertainty in Artificial Intelligence (UAI'95), Montreal, Quebec, Canada, 18–20 August, 1995. – SF.: Morgan Kaufmann, 1995. – P. 296-305.
205. Neil M., Fenton N., Forey S. and Harris R. Using Bayesian belief networks to predict the reliability of military vehicles. [Text] / Computing and Control Engineering Journal. – 2001. – Vol.2. – P.11-20.
206. Horvitz E., Ruokangas C. and Srinivas S. A decision-theoretic approach to the display of information for time-critical decisions: the Vista Project. [Text] / In Proceedings of annual workshop on Space Operations, Applications and Research (SOAR'92), NASA/Johnson Space Center, Houston, Texas, US, 4-6 August, 1992. – NY.: Elsevier science, 1992. – P. 407-417.
207. Horvitz E. and Barry M. Display of information for time-critical decision making. [Text] / Proceedings of the eleventh annual conference on Uncertainty in Artificial Intelligence (UAI'95), Montreal, Quebec, Canada, 18–20 August, 1995. – SF.: Morgan Kaufmann, 1995. – P. 296-305.
208. Fung R. and Favero B. Applying Bayesian Networks to information retrieval // Communications of the ACM (CACM). – 1995. – Vol. 38, № 3. – P. 24-30.
209. L. M. de Campos, Fernandez-Luna J. M., Huete J. F., Martin-Dancausa C., Tagua-Jimenez A., Tur-Vigil C. An integrated system for managing the Andalusian parliament's digital library. [Text] / Program-Electronic Library and Information Systems 43(2): 156-174, 2009.
210. Lo B., Thiemjarus S. and Yang G. Adaptive Bayesian networks for video processing / IEEE International conference on image processing (ICIP 2003), Barcelona (Spain), 14–17 September, 2003. – 1. – P. 889-892.
211. Pham T.V., M. Worrying, A.W. Smeulders. Face Detection by Aggregated Bayesian Network Classifiers. [Text] / Pattern Recognition Letters. – 2002. - Vol. 23, №4. – P.451-461.
212. Bilnes J.A. Dynamic Bayesian Multinets. [Text] / Proceedings of the 16th Annual Conference on Uncertainty in AI. – 2000. – P.38-45.
213. Nefian A.V., Liang L.H., Pi X.B., Liu X.X., and Murphy K. Dynamic Bayesian Networks for Audio-Visual Speech Recognition. [Text] / Eurasip Journal on Advances in Signal Processing. – 2002. – Vol.11. – P.1274-1288.
214. Офіційний сайт Вашингтонського університету [Електронний ресурс]. – Режим доступу : [www.ssli.ee.washington.edu/~bilmes](http://www.ssli.ee.washington.edu/~bilmes),
215. Офіційний сайт Intel [Електронний ресурс]. – Режим доступу : [www.intel.com/research/mrl/research/avcsr.htm](http://www.intel.com/research/mrl/research/avcsr.htm).
216. Терентьев А.Н. SAS BASE: Основы программирования (научное издание) / Терентьев А.Н., Домрачев В.Н., Костецкий Р.И. – К: Эдельвейс, 2014. – 304 с. – ISBN 978-966-2748-49-9
217. D. Dean, J. Wexler. What's New in SAS Enterprise Miner 13.1 // Technical report SAS311-2014. – SAS Institute Inc., Cary, NC, USA, 2014. – 16 p.
218. SAS Enterprise Miner 13.2: Reference Help. SAS Documentation. – SAS Institute Inc., Cary, NC, USA, 2014. – 1740 p.

219. Yongqiao Xiao. SAS Enterprise Miner 13.2 High-Performance Procedures. – SAS Institute Inc., Cary, NC, USA, 2014. – 285 p.
220. J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. – San Francisco: Morgan Kaufmann, 1988. – 552 p.
221. Horvitz E. and Barry M. Display of information for time-critical decision making. [Text] / Proceedings of the eleventh annual conference on Uncertainty in Artificial Intelligence (UAI'95), Montreal, Quebec, Canada, 18–20 August, 1995. – SF.: Morgan Kaufmann, 1995. – P. 296-305.
222. Horvitz E., Ruokangas C. and Srinivas S. A decision-theoretic approach to the display of information for time-critical decisions: the Vista Project. [Text] / In Proceedings of annual workshop on Space Operations, Applications and Research (SOAR'92), NASA/Johnson Space Center, Houston, Texas, US, 4-6 August, 1992. – NY.: Elsevier science, 1992. – P. 407-417.
223. Horvitz E. and Barry M. Display of information for time-critical decision making. [Text] / Proceedings of the eleventh annual conference on Uncertainty in Artificial Intelligence (UAI'95), Montreal, Quebec, Canada, 18–20 August, 1995. – SF.: Morgan Kaufmann, 1995. – P. 296-305.
224. Терентьев А.Н. Алгоритм вероятностного вывода в байесовских сетях / А.Н. Терентьев, П.И. Бидюк, Л.А. Коршевич / Системный анализ и информационный технологии: сб. науч. трудов по материалам IX междунар. науч.-тех. конф., 15–19 мая 2007 г., Киев. – К.: Екмо, 2007. – С. 76.
225. Spirtes P., Glymour C. and Scheines R. Causality from probability / Proceedings of advanced computing for the social sciences, Williamsburgh. – 1990. – P. 107-121.
226. Бидюк П. І. Застосування байєсівського підходу в медичній діагностиці / П. І. Бидюк, О. М. Терентьев / Автоматика-2004: зб. наук. праць за матеріалами XI міжнар. конф., 27–30 вер. 2004 р., Київ. – К.: НУХТ, 2004. – Т. 3 – С. 32.
227. Терентьев А. Н. Использование байесовских сетей в кредитном скоринге / А. Н. Терентьев, П. И. Бидюк / Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта: сб. науч. трудов по материалам междунар. конф., 19 мая-23 мая 2008 г., Евпатория. – Херсон: ПП Вишемирский В. С., 2008. – Т. 3, №2 – С. 101-103.
228. Терентьев О.М. Застосування SPI-методу для формування ймовірного висновку в мережах Байєса / О.М. Терентьев, П.І. Бидюк / Математичне та програмне забезпечення інтелектуальних систем: зб. наук. праць за матеріалами V міжнар. наук.-прак. конф., 14-16 лист. 2007 р., Дніпропетровськ. – Дніпропетровськ: ДНУ, 2007. – С. 189-190.
229. Heckerman D. A Tutorial on Learning with Bayesian Networks [Text] / Learning in Graphical Models. – Cambridge: MIT Press, 1998. – P. 301-354.
230. Згуровський М. З. Основи системного аналізу [Текст] / М. З. Згуровський, Н. Д. Панкратова. – К.: Видавнича група ВНУ, 2004. – 544 с.
231. Спицнадель В. Н. Основы системного анализа [Текст] : учебн. пособие / В. Н. Спицнадель. – СПб.: Изд. дом "Бизнес-пресса", 2000. – 204 с.
232. Turban E. Decision support and expert systems: management support systems [Text] / Efraim Turban. - N.J.: Prentice Hall, 1995. – 960 p.
233. Neapolitan R.E. Probabilistic Reasoning in Expert Systems: Theory and Algorithms [Text] / Richard E. Neapolitan. – Wiley, NY, 1989. – 448 p.

234. Lauritzen S. L. The EM algorithm for graphical association models with missing data [Text] / Computational Statistics and Data Analysis. – 1995. – Vol. 19. – P. 191-201
235. Madsen A. L. and Jensen F. V. Lazy propagation in junction trees / Fourteenth conference on Uncertainty in Artificial Intelligence (UAI'98), Madison, Wisconsin, USA, 24–26 July, 1998. – SF.: Morgan Kaufmann, 1998. – P. 362-369.
236. Chavez R. M. and Cooper G. F. A randomized approximation algorithm for probabilistic inference on Bayesian belief networks // Networks. – Wilmington: Wiley Periodicals Inc, 1990. – Vol. 20, № 5. – P. 661-685.
237. Druzdzel M. J. Some properties of joint probability distributions / Proceedings of the tenth annual conference on Uncertainty in Artificial Intelligence (UAI'94), Seattle, Washington, USA, 29–31 July, 1994. – SF.: Morgan Kaufmann, 1994. – P.187-194.
238. Pearl J. Causal inference in statistics: An overview // Statistics Surveys. – 2009. – 3. – P. 96 – 146.

ДОДАТКИ

Додаток А

Програмне забезпечення: назва, доступ в мережі, розробники

Назва	Доступ в мережі	Розробники
Analytica	<a href="http://www.lumina.com">http://www.lumina.com</a>	Lumina (Henrion)
Bassist	<a href="http://www.cs.Helsinki.FI/research/fdk/bassist">http://www.cs.Helsinki.FI/research/fdk/bassist</a>	U. Helsinki
Bayda	<a href="http://www.cs.Helsinki.FI/research/cosco/Projects/NOE/SW">http://www.cs.Helsinki.FI/research/cosco/Projects/NOE/SW</a>	U. Helsinki
Bayes Builder	<a href="http://www.mbfys.kun.nl/snn/Research/bayesbuilder/">http://www.mbfys.kun.nl/snn/Research/bayesbuilder/</a>	Nijman (U. Nijmegen)
Bayesia Lab	<a href="http://www.bayesia.com">http://www.bayesia.com</a>	Bayesia Ltd
Bayesware Discoverer	<a href="http://www.bayesware.com">http://www.bayesware.com</a>	Bayesware (Open Univ., UK)
B-course	<a href="http://b-course.cs.helsinki.fi">http://b-course.cs.helsinki.fi</a>	U. Helsinki
BN power constructor	<a href="http://www.cs.ualberta.ca/~jcheng/bnpc.htm">http://www.cs.ualberta.ca/~jcheng/bnpc.htm</a>	Cheng (U. Alberta)
BNet Master	mail to: <a href="mailto:o.terentiev@gmail.com">o.terentiev@gmail.com</a>	Trofymenko, Davydenko, Terentiev, (NTUU KPI, Kyiv, Ukraine)
BNT	<a href="http://www.ai.mit.edu/~murphyk/Software/BNT/bnt.html">http://www.ai.mit.edu/~murphyk/Software/BNT/bnt.html</a>	Murphy (prev. U.C. Berkeley, now MIT)
BNJ	<a href="http://bndev.sourceforge.net/">http://bndev.sourceforge.net/</a>	Hsu (Kansas)
Bucket Elim	<a href="http://www.ics.uci.edu/~irinar">http://www.ics.uci.edu/~irinar</a>	Rish (U.C. Irvine)
BUGS	<a href="http://www.mrc-bsu.cam.ac.uk/bugs">http://www.mrc-bsu.cam.ac.uk/bugs</a>	MRC/Imperial College
Business Navigator 5	<a href="http://www.data-digest.com">http://www.data-digest.com</a>	DataDigestCorp
CABeN	<a href="http://www-pcd.stanford.edu/cousins/caben-1.1.tar.gz">http://www-pcd.stanford.edu/cousins/caben-1.1.tar.gz</a>	Cousins et al. (Wash. U.)
CaMML	<a href="http://www.datamining.monash.edu.au/software/camml">http://www.datamining.monash.edu.au/software/camml</a>	Wallace, Korb (Monash U.)
CoCo+Xlisp	<a href="http://www.math.auc.dk/~jhb/CoCo/information.html">http://www.math.auc.dk/~jhb/CoCo/information.html</a>	Badsberg (U. Aalborg)
CIspace	<a href="http://www.cs.ubc.ca/labs/lci/CIspace/">http://www.cs.ubc.ca/labs/lci/CIspace/</a>	Poole et al. (UBC)
Deal	<a href="http://www.math.auc.dk/novo/deal">http://www.math.auc.dk/novo/deal</a>	Bottcher et al.
Ergo	<a href="http://www.noeticsystems.com">http://www.noeticsystems.com</a>	Noetic systems
GDAGsim	<a href="http://www.staff.ncl.ac.uk/d.j.wilkinson/software/gdagsim/">http://www.staff.ncl.ac.uk/d.j.wilkinson/software/gdagsim/</a>	Wilkinson (U. Newcastle)
GMRFSim	<a href="http://www.math.ntnu.no/~hrue/GMRFSim/">http://www.math.ntnu.no/~hrue/GMRFSim/</a>	Rue (U. Trondheim)
GeNIe/SMILE	<a href="http://www.sis.pitt.edu/~genie">http://www.sis.pitt.edu/~genie</a>	U. Pittsburgh (Druzdzel)

Назва	Доступ в мережі	Розробники
GMTk	<a href="http://ssli.ee.washington.edu/~bilmes/gmtk/">http://ssli.ee.washington.edu/~bilmes/gmtk/</a>	Bilmes (UW), Zweig (IBM)
gR	<a href="http://www.r-project.org/gR">http://www.r-project.org/gR</a>	Lauritzen et al.
Grappa	<a href="http://www.stats.bris.ac.uk/~peter/Grappa/">http://www.stats.bris.ac.uk/~peter/Grappa/</a>	Green (Bristol)
Hugin	<a href="http://www.hugin.com">http://www.hugin.com</a>	HuginExpert (U. Aalborg, Lauritzen/Jensen)
Hydra	<a href="http://software.biostat.washington.edu/statsoft/MC/MC/Hydra">http://software.biostat.washington.edu/statsoft/MC/MC/Hydra</a>	Warnes (U.Wash.)
Ideal	<a href="http://yoda.cis.temple.edu:8080/ideal/">http://yoda.cis.temple.edu:8080/ideal/</a>	Rockwell (Srinivas)
JavaBayes	<a href="http://www.cs.cmu.edu/~javabayes/Home/">http://www.cs.cmu.edu/~javabayes/Home/</a>	Cozman (CMU)
MIM	<a href="http://www.hypergraph.dk/">http://www.hypergraph.dk/</a>	HyperGraph Software
MSBNx	<a href="http://research.microsoft.com/adapt/MSBNx/">http://research.microsoft.com/adapt/MSBNx/</a>	Microsoft
Netica	<a href="http://www.norsys.com">http://www.norsys.com</a>	Norsys (Boerlage)
PMT	<a href="http://people.bu.edu/vladimir/pmt/index.html">http://people.bu.edu/vladimir/pmt/index.html</a>	Pavlovic (BU)
PNL	<a href="http://www.ai.mit.edu/murphyk/Software/PNL/pnl.html">http://www.ai.mit.edu/murphyk/Software/PNL/pnl.html</a>	Eruhimov (Intel)
Pulcinella	<a href="http://iridia.ulb.ac.be/pulcinella/Welcome.html">http://iridia.ulb.ac.be/pulcinella/Welcome.html</a>	IRIDIA
RISO	<a href="http://sourceforge.net/projects/riso">http://sourceforge.net/projects/riso</a>	Dodier (U. Colorado)
TETRAD	<a href="http://www.phil.cmu.edu/tetrad/">http://www.phil.cmu.edu/tetrad/</a>	CMU
UnBBayes	<a href="https://sourceforge.net/projects/unbbayes/">https://sourceforge.net/projects/unbbayes/</a>	?
Vibes	<a href="http://www.inference.phy.cam.ac.uk/jmw39/">http://www.inference.phy.cam.ac.uk/jmw39/</a>	Winn & Bishop (U. Cambridge)
Web Weaver	<a href="http://snowwhite.cis.uoguelph.ca/faculty/info/yxiang/ww3/">http://snowwhite.cis.uoguelph.ca/faculty/info/yxiang/ww3/</a>	Xiang (U.Regina)
WinMine	<a href="http://research.microsoft.com/~dmax/WinMine/tooldoc.htm">http://research.microsoft.com/~dmax/WinMine/tooldoc.htm</a>	Microsoft
XBAIES 2.0	<a href="http://www.staff.city.ac.uk/~rgc/webpages/xbpage.html">http://www.staff.city.ac.uk/~rgc/webpages/xbpage.html</a>	Cowell (City U.)

## STATISTICAL ANALYSIS SYSTEM – СУЧАСНИЙ ІНСТРУМЕНТ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

### *Історичні відомості про систему SAS*

SAS Institute – американська приватна компанія, що займається розробкою програмного забезпечення і засобів бізнес-аналітики. Компанія була заснована в 1976 році Ентоні Барром (Anthony Barr), Джеймсом Гуднайтом (James Goodnight), Джоном Соллом (John Sall) і Джейн Хельвінг (Jane Helving). Свою назву SAS придбала від скорочення повної назви Statistical Analysis System, хоча станом на даний момент продукти компанії вже вийшли далеко за межі статистичного аналізу.

Компанія SAS Institute починалася як науково-дослідний проект з аналізу даних у сільському господарстві на базі університету штату Північної Кароліни. У 1966 році Національний Інститут Здоров'я (англ. National Institute of Health, NIH) виділив грант з метою розробки універсального статистичного пакету прикладних програм, призначеного для аналізу даних сільського господарства. У проекті взяли участь вісім університетів, лідером серед яких був NCSU (North Carolina State University at Raleigh). Спеціально був створений новий орган – Університет статистики Південних Експериментальних Станцій (The University Statisticians of the Southern Experiment Stations, USSSES), під патронатом Міністерства сільського господарства США. Не останню роль у проекті зіграла компанія IBM, для якої цей проект був вкрай вигідний, завдяки розповсюдженню комп'ютерів **IBM System / 360**. Програмне забезпечення цієї серії комп'ютерів потребувала оновлення, тому було пов'язане з апаратною частиною, як наслідок програми незалежні від “заліза” було вкрай потрібне, для збільшення продаж нових моделей [1].

### *Корпоративна статистика компанії SAS Institute*

На початок 2015 рік компанія має більше 400 офісів в яких працює більше 13 660 співробітників, які підтримують роботу понад 75 000 компаній в 139 країнах світу.

Співробітники компанії SAS розміщені по країнах світу наступним чином:

- на території США працює 6964 співробітників, з яких 5133 в штаб-квартирі в м Керрі, штат Північна Кароліна;
- Канаді – 307 співробітників;
- у Латинська Америка – 458 співробітників.
- Європа, Середній Схід і Африка – 3701 співробітників.
- Азіатсько-Тихоокеанський регіон – 2230 співробітників.

У 2014 році дохід компанії склав 3,09 млрд. \$, що на 2,3% більше, ніж у попередньому 2013 році. Завдяки тому, що компанія є приватною, тобто вона не проводила ІРО, як наслідок немає необхідності виплачувати дивіденди акціонерам, 25% щорічного прибутку компанії спрямовується на дослідження і розробки. В цілому, цей показник по галузі в інших компаніях в два рази менше. Станом на 2014 рік компанією представлено понад 200 рішень і продуктів загального використання, а також спеціалізованих для відповідних галузей.

Таблиця 1

Зміна доходу компанії SAS Institute з 1976 по 2014 роки

Рік	Дохід	Рік	Дохід	Рік	Дохід
1976	138 тис.	1989	205.6 млн.	2002	1.18 млрд.
1977	506 тис.	1990	240.2 млн.	2003	1.34 млрд.
1978	1.2 млн.	1991	295.4 млн.	2004	1.53 млрд.
1979	2.3 млн.	1992	365.5 млн.	2005	1.68 млрд.
1980	5.05 млн.	1993	420.2 млн.	2006	1.90 млрд.
1981	11.2 млн.	1994	481.9 млн.	2007	2.15 млрд.
1982	19.6 млн.	1995	562.4 млн.	2008	2.26 млрд.
1983	32.3 млн.	1996	652.8 млн.	2009	2.31 млрд.
1984	52.6 млн.	1997	750.0 млн.	2010	2.43 млрд.
1985	70.9 млн.	1998	871.4 млн.	2011	2.75 млрд.
1986	102.4 млн.	1999	1.02 млрд.	2012	2.87 млрд.
1987	135.3 млн.	2000	1.12 млрд.	2013	3.02 млрд.
1988	170.4 млн.	2001	1.13 млрд.	2014	3.09 млрд.

Перший офіс компанії на пострадянському просторі був відкритий у Москві в 1996 році, а в 2007 році в Україні. Із загального обсягу продажів у країнах Центральної та Східної Європи 34% припадає на СНГ. За обсягом продажів лідерами є телекомунікаційний сектор (44%), банки та страхові компанії (29%), транспорт (15%), 12% приходить на всі інші сектори економіки.

В Україні програмне забезпечення SAS забезпечує аналітичну обробку даних в ТОП-30 банківських установ, телекомунікаційна компанія МТС, міністерстві внутрішніх справ, Метінвест та багато інших.

### ***Огляд програмних продуктів компанії SAS.***

Одними з найбільш популярних продуктів компанії SAS є **SAS Analytics Pro** що складається з модулів SAS Base, SAS / STAT та SAS / GRAPH, що поєднують простоту налаштування з можливістю використання на одному персональному комп'ютері. Разом вони утворюють комплексний інструмент для аналітиків, дослідників, статистиків, інженерів і вчених.

SAS Analytics Pro забезпечує доступ користувачів до інтегрованого і масштабованого програмного середовища для перегляду і трансформації даних, зберігання та пошуку інформації, статистичного аналізу даних, складання звітів і візуалізації результатів. Окрім цього, потужний набір макросів зменшує час розробки програм і їх обслуговування. Все це дозволяє ІТ службам скоротити та пришвидшити програмування, знизити витрати і отримати в результаті аналітичну інформацію і звіти, необхідні особам, що відповідають за прийняття рішень [2].

Програмне забезпечення SAS інтегрується в будь-яке комп'ютерне середовище, що дозволяє об'єднати обчислювальні можливості компанії і отримати єдину картину даних організації. Аналітики можуть зосередитися на аналізі даних, а не витрачати час на вирішення проблем з ними пов'язаними.

Програмне забезпечення SAS може перекрити всі потреби організації при графічному відображенні даних, включаючи аналітичні графіки, карти і діаграми з будь-яким необхідним стилем відображення. Користувачі можуть з легкістю

створювати звітність в стандартних офісних форматах, таких як Rich Text Format (RTF) та Portable Document Format (PDF), отримуючи на виході документи з будь-яким набором мов розмітки, що підвищує їх продуктивність і знижує навантаження на ІТ департамент.

**SAS Enterprise Intelligence Platform** – найважливіший комплекс інструментів інтеграції, управління та аналізу даних і подальшого надання отриманої інформації для використання в рамках всього підприємства.

**SAS Intelligence Solutions** – програмні модулі, розроблені спеціально для вирішення найважливіших проблем конкретних напрямів і функціональних областей бізнесу, а саме – маркетингу, фінансів, ІТ, управління персоналом та постачання.

**SAS Industry Solutions** – програмне забезпечення, призначене для відповідних галузей. Воно містить необхідні спеціалізовані знання і моделі даних, створені, щоб допомогти компаніям прискорити досягнення своїх цілей, одночасно знизивши ризики і витрати при впровадженні.

**Середовище програмування SAS Base.** Призначене для управління даними і виклику спеціалізованих статистичних процедур, також можуть використовуватися процедури SQL (Structured Query Language). Реалізовані механізми доступу до даних надають можливість роботи з різноманітними структурами даними, у тому числі і з Oracle. Є можливість створення звітів у форматах RTF, PDF, XML, HTML і звичайному текстовому файлі. Середа SAS реалізована у вигляді вікон інтерактивного графічного інтерфейсу користувача, призначеного для запуску і тестування sas-програм. SAS Base є основою розроблених компанією продуктів. Зв'язок SAS Base з іншими модулями показана на рис. 1.



Рис. 1. Місце SAS Base серед продуктів компанії.

Окрім SAS Base компанія пропонує багато інших програмних продуктів. Нижче наведено короткий опис декількох рішень і програм, розроблених компанією. В цілому компанія пропонує більше 200 продуктів для різних галузей, призначених для зберігання, обробки та аналізу даних. Описані нижче програмні продукти обрані авторами суб'єктивно на підставі популярності та перспективності в Україні.

**SAS Enterprise Guide** – програма, в якій реалізована технологія Drag and Drop розміщення різноманітних sas-процедур і функцій для побудови аналітичного



процесу у вигляді технологічного процесу. Ця програма містить в собі процедури і функції SAS Base, що можна використовувати у вигляді sas-програм.

**SAS / GRAPH** – програмне забезпечення для візуалізації даних, що використовується аналітиками та ІТ керівниками для розуміння процесів. Входить до складу багатьох індустріальних рішень, а також SAS Enterprise Guide.

**SAS / ETS** (Econometrics and Time-Series analysis) пропонує широкий спектр інструментів: аналізу часових рядів та економетричних методів прогнозування. Ці інструменти дозволяють аналізувати, прогнозувати і моделювати процеси бізнесу для поліпшення стратегічного і тактичного планування. Цей інструмент допомагає краще зрозуміти такі фактори, як економічні та ринкові умови, демографія, які впливають на цінові рішення і маркетингові заходи в бізнесі.

**SAS / GIS** (Geographic Information System) інструмент для інтерактивного дослідження географічної інформації, що дозволяє систематизувати і аналізувати дані, які можна пов'язати з фізичним місцем розташування. Використовується для демографічних, маркетингових та епідеміологічних досліджень.

**SAS / IML** (Interactive Matrix Language) – програмне забезпечення, в якому реалізована спеціалізована мова програмування IML, для роботи з матрицями в динамічному середовищі. В першу чергу мова програмування IML призначена для програмістів, аналітиків-статистиків і дослідників. Підтримується робота з R-кодом. Дуже розповсюджено в західних навчальних закладах поряд з такими системами як Matlab, MathCad та R.

**SAS / OR** (operation research) програмне забезпечення для вирішення завдань оптимізації, моделювання та календарного планування, для визначення дій, що забезпечують найкращі результати, в умовах обмеження ресурсів і часу. Це дозволяє організаціям розглядати спектр альтернативних дій і сценаріїв, краще керувати ресурсами та розробляти плани для досягнення поставленої мети.

**SAS / STAT** (Statistical Analysis) призначений як для звичайних так і корпоративних аналітичних служб. Містить як традиційні методи статистичного аналізу даних – дисперсійного аналізу і прогнозного моделювання, так і спеціалізовані – байєсівський, категоріальний, багатовимірний, живучості, психометричний, кластерний, непараметричний та інші типи аналізу.

**SAS Enterprise Miner** – спеціалізований інструмент призначений для спрощення процесу аналізу даних, при створенні високоточних інтелектуальних і описових моделей, заснованих на великих обсягах даних в масштабі всього підприємства. Інструмент містить широкий набір методів інтелектуального аналізу даних – регресійний аналіз, дерева рішень, нейронні мережі, машина опорних векторів, кластерний аналіз та багато інших спеціалізованих найсучасніших методів. Інструмент використовується для вирішення завдань виявлення шахрайства, мінімізації ризиків, прогнозування потреби в ресурсах, скорочення часу простою активів, збільшення ефективності маркетингових кампаній, запобігання відтоку клієнтів, розробки скорінгових карт, збільшення продажів товарів та іншого.

З 2010 року Науково-навчальний комплекс “Інститут Прикладного Системного Аналізу” Національного Технічного Університету України “Київський Політехнічний Інститут” (НТУУ КПІ) бере участь у Глобальній Академічній Програмі SAS, сприяє використанню програмного забезпечення SAS у навчанні, викладанні та наукових дослідженнях.

Продовженням співпраці ННК ІІСА НТУУ КІІ та SAS стало відкриття на його базі у лютому 2012 року Центру Компетенції SAS, задачами якого є:

- надання консультаційних послуг при впровадженні системи SAS в державних та комерційних установах;
- підготовка методичних навчальних матеріалів;
- навчання студентів та фахівців роботі з системою та галузевими продуктами SAS;
- проведення досліджень в різноманітних предметних галузях на основі використання аналітичних і статистичних підходів, з метою виявлення закономірностей, побудови математичних моделей, їх впровадженням та подальшим моніторингом;
- розробка нових методів статистичного та інтелектуального аналізу даних.

Доступ до програмного забезпечення SAS на пільгових умовах дозволив доповнити вже існуючі курси навчальних дисциплін практично значущими лабораторними роботами, а також розробити нові навчальні курси. Отримані знання надають можливість випускникам працювати в організаціях, де використовується система, а це майже половина українських банків зі списку TOP 30, телекомунікаційна компанія МТС, страхові компанії та інші.

### Література

1. Терентьев А.Н. SAS BASE: Основы программирования (научное издание) / Терентьев А.Н., Домрачев В.Н., Костецкий Р.И. – К: Эдельвейс, 2014. – 304 с. – ISBN 978-966-2748-49-9
2. D. Dean, J. Wexler. What's New in SAS Enterprise Miner 13.1 // Technical report SAS311-2014. – SAS Institute Inc., Cary, NC, USA, 2014. – 16 p.
3. SAS Enterprise Miner 13.2: Reference Help. SAS Documentation. – SAS Institute Inc., Cary, NC, USA, 2014. – 1740 p.
4. Yongqiao Xiao. SAS Enterprise Miner13.2 High-Performance Procedures. – SAS Institute Inc., Cary, NC, USA, 2014. – 285 p.
5. J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. – San Francisco: Morgan Kaufmann, 1988. – 552 p.

Опис функціональних можливостей програмного забезпечення для БМ

Назва	Функціональні можливості	Характеристики
Src	Вихідний код програми	N= немає J = Java; M = Matlab; D = Delphi; L = Lisp C, C++, R – мови програмування
API	Інтерфейс прикладних програм	N = програма не може бути об'єднана з вашим кодом, тобто має виконуватись як автономна програма Y = програма може бути об'єднана
Exec	Операційна система	W = Windows (95/98/200/NT) U = Unix M = MacIntosh - = будь яка з компілятором
GUI	Графічний інтерфейс користувача	N = ні Y = так
D/C	Неперервність/або дискретні типи вершин	D = лише дискретні вершини G = гаусівські вершини (аналітично) Cs = неперервні вершини (вибірка) Cd = неперервні вершини (дискретизація) Cx = неперервні вершини (невідомий метод)
Param	Параметричне навчання	N = ні Y = так
Struct	Структурне навчання	N= Ні Y= Так C1 = тести на умовну незалежність K2 = алгоритм K2 Купера і Герсковіца
D/U	Типи графів	U = лише неорієнтовані графи D = лише орієнтовані графи UD = орієнтовані та неорієнтовані графи CG = ланцюгові графи (змішані орієнтовані та неорієнтовані)
Infer	Алгоритм ймовірнісного виводу	JT = зв'язне дерево (Junction Tree) VE = виключення змінних PT = Pearl's polytree E = точний MH = Metropolis-Hastings MC = Markov Chain Monte Carlo (MCMC) GS = Gibbs sampling IS = вибірка за значущістю S = формування вибірки O = інший ++ = підтримка багатьох методів ? = не визначений N = немає (+S) = точний з формуванням вибірки
Free	Безкоштовна версія програми	O = Так (навіть якщо лише з навчальною метою) \$ = Ні, комерційна (+обмежені версії)

**Навчальне видання**

*Навчальний посібник*

М. З. Згуровський, П. І. Бідюк, О. М. Терентьев, Т. І. Просянкіна-Жарова

**БАЙЄСІВСЬКІ МЕРЕЖІ В СИСТЕМАХ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ**

Підписано до друку 30.06.2015 р. Формат 60x84/8.  
Папір офсетний. Надруковано на ризографі.  
Умовн. друк. арк.37,5. Облік.-вид. арк. 34,88.  
Тираж 100 прим. Зам. № 037.

Свідоцтво про державну реєстрацію суб'єкта видавничої діяльності  
серія ДК № 4249 від 29.12.2011 р.

Друк: ТОВ «Видавниче Підприємство «Едельвейс»  
03170, м. Київ, вул. Зодчих, 74, тел.: (044)361-78-68.